

**UNIVERSIDAD DE CONCEPCIÓN - CHILE
FACULTAD DE INGENIERIA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL**

**Algoritmo GRASP para la Programación de Cirugías
Electivas en un Hospital Público Chileno**

por

Ignacio Isaías Cartes Rubilar

Profesor guía

Dra. Rosa Medina Durán

Concepción, Abril de 2016

Tesis presentada a la

**DIRECCIÓN DE POSTGRADO
UNIVERSIDAD DE CONCEPCIÓN**



Para optar al grado de

MAGISTER EN INGENIERIA INDUSTRIAL

This study was partially supported by ECOS/CONICYT C13E04



Dedicatoria

A mis padres, que siempre me han dado libertad y guiado para hacer lo que me propongo.

A mi esposa, que es la compañera de mi vida, que me ayudó con esta tesis, me subía el ánimo y me aconsejaba cada vez que lo necesitaba.

A mis amigos Diego Soto y Daniel Seguel, por su buena voluntad y disposición, por abrirme las puertas de sus casas y ayudarme con tardes y noches de programación.

RESUMEN

ALGORITMO GRASP PARA LA PROGRAMACIÓN DE CIRUGÍAS ELECTIVAS EN UN HOSPITAL PÚBLICO CHILENO

Ignacio Isaías Cartes Rubilar

Abril de 2016

PROFESOR GUÍA: Rosa Medina Durán

PROGRAMA: Magíster en Ingeniería Industrial

Los hospitales en Chile, reaccionando a la gran demanda actual y la necesidad de entregar servicios con recursos limitados (entre algunas cosas), han debido buscar métodos para la optimización de los recursos.

En este estudio, además de presentar un estudio del estado del arte de la programación de cirugías electivas, se realiza una comparación en la resolución del problema de programación de cirugías electivas de un modelo IP presentado en un trabajo anterior con un algoritmo basado en la metaheurística *Greedy Randomized Adaptive Search Procedures* (GRASP) mediante instancias obtenidas con datos históricos de un hospital público chileno.

El tiempo de ejecución del algoritmo GRASP es considerablemente menor al tiempo de ejecución del modelo lineal entero, siendo para la instancia de mayor tamaño 130800 segundos de ejecución versus 1.99 segundos del algoritmo GRASP, logrando una calidad de solución similar. Se resuelven además, instancias de mayor tamaño con el algoritmo GRASP considerando hasta 250 pacientes y 22 médicos.

Palabras Claves: *Scheduling*, GRASP, salas quirúrgicas.

ABSTRACT

A GRASP ALGORITHM FOR THE ELECTIVE SURGERIES SCHEDULING PROBLEM IN A CHILEAN PUBLIC HOSPITAL

Ignacio Isaías Cartes Rubilar

April, 2016

THESIS SUPERVISOR: Rosa Medina Durán

PROGRAM: Master in Industrial Engineering

Hospitals in Chile, reacting to both high current demand and the need to deliver services with limited resources, have had to search different methods for resources optimization.

This study, besides presenting a state of the art of elective surgeries scheduling, makes a comparison between solving the elective surgeries scheduling problem with an Integer Programming model presented in an earlier study and an algorithm based on the Greedy Randomized Adaptive Search Procedures (GRASP) metaheuristic using scenarios taken from historical data of a Chilean public hospital.

The GRASP algorithm considerably improves the execution time obtained by the Integer Programming model, with a time of 1.99 seconds in the biggest scenario, compared to the 130800 seconds obtained with IP, and including a similar quality of solution. Thus, the GRASP algorithm can solve different scenarios, solving a 250 patients and 22 doctors scenario as its biggest scenario.

Keywords: Scheduling, operating rooms, GRASP.

Tabla de Contenidos

| | |
|---|-----|
| LISTA DE TABLAS | IX |
| LISTA DE FIGURAS..... | XI |
| ABREVIACIONES | XII |
| CAPÍTULO 1. INTRODUCCIÓN..... | 1 |
| 1.1. INTRODUCCIÓN GENERAL | 1 |
| 1.2. HIPÓTESIS DE TRABAJO..... | 2 |
| 1.3. ALCANCES Y LIMITACIONES | 2 |
| 1.4. OBJETIVOS..... | 3 |
| 1.4.1 <i>Objetivo general</i> | 3 |
| 1.4.2 <i>Objetivos específicos</i> | 3 |
| 1.5. METODOLOGÍA | 4 |
| CAPÍTULO 2. DESCRIPCIÓN DEL PROBLEMA..... | 6 |
| 2.1. INTRODUCCIÓN..... | 6 |
| 2.2. CARACTERÍSTICAS DE LOS PACIENTES | 6 |
| 2.3. CARACTERÍSTICAS DE LA UNIDAD DE PABELLÓN | 7 |
| 2.4. CARACTERÍSTICAS DE LOS MÉDICOS | 8 |
| 2.5. CARACTERÍSTICAS DE LAS SALAS QUIRÚRGICAS..... | 8 |
| 2.6. CARACTERÍSTICAS DEL EQUIPO MÉDICO | 9 |
| 2.7. PLANIFICACIÓN Y PROGRAMACIÓN DE CIRUGÍAS | 9 |
| 2.8. MODELOS Y ESTRATEGIAS DE PROGRAMACIÓN | 10 |
| 2.9. PRIORIZACIÓN DE PACIENTES ELECTIVOS | 11 |
| CAPÍTULO 3. MARCO TEÓRICO | 12 |
| 3.1. INTRODUCCIÓN..... | 12 |

| | | |
|--|---|----|
| 3.2. | SCHEDULLING (PINEDO, 2008) | 12 |
| 3.3. | PROBLEMAS DE ASIGNACIÓN (BURKARD, DELL'AMICO, & MARTELLO, 2009) | 16 |
| 3.3.1 | <i>Problemas de asignaciones lineales</i> | 16 |
| 3.4. | OPTIMIZACIÓN MULTIOBJETIVO (MARLER & ARORA, 2004) | 18 |
| 3.4.1 | <i>Definición del problema de optimización multiobjetivo</i> | 18 |
| 3.4.2 | <i>Optimalidad de Pareto</i> | 19 |
| 3.4.3 | <i>Métodos de optimización multiobjetivo</i> | 19 |
| 3.5. | METAHEURÍSTICAS (GOGNA & TAYAL, 2013)..... | 20 |
| 3.6. | <i>GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURES</i> (RESENDE & RIBEIRO, 2003) | 21 |
| 3.7. | MÉTODOS DE RESOLUCIÓN | 23 |
| CAPÍTULO 4. ESTADO DEL ARTE | | 24 |
| 4.1. | INTRODUCCIÓN..... | 24 |
| 4.2. | REVISIÓN BIBLIOGRÁFICA | 24 |
| 4.2.1 | <i>Trabajos locales</i> | 24 |
| 4.2.2 | <i>Priorización de pacientes</i> | 27 |
| 4.3. | TRABAJOS INTERNACIONALES | 28 |
| 4.4. | ANÁLISIS DE LA REVISIÓN BIBLIOGRÁFICA DEL PROBLEMA | 34 |
| CAPÍTULO 5. MODELAMIENTO DEL PROBLEMA DE CIRUGÍAS ELECTIVAS..... | | 39 |
| 5.1. | INTRODUCCIÓN..... | 39 |
| 5.2. | CONSIDERACIONES PARTICULARES..... | 39 |
| 5.3. | MODELO DE PROGRAMACIÓN LINEAL ENTERO..... | 40 |
| 5.3.1 | <i>Índices</i> | 40 |
| 5.3.2 | <i>Parámetros</i> | 40 |
| 5.3.3 | <i>Variables de decisión</i> | 40 |
| 5.3.4 | <i>Funciones objetivo</i> | 41 |
| 5.3.5 | <i>Restricciones del modelo</i> | 41 |

| | | |
|---|---|----|
| 5.4. | IMPLEMENTACIÓN | 44 |
| CAPÍTULO 6. ALGORITMO GRASP PARA EL PROBLEMA DE PROGRAMACIÓN DE CIRUGÍAS ELECTIVAS Y EXPERIMENTOS | | |
| 6.1. | INTRODUCCIÓN..... | 45 |
| 6.2. | ALGORITMO GRASP | 46 |
| 6.2.1 | <i>Representación de la solución</i> | 46 |
| 6.2.2 | <i>Función Fitness</i> | 47 |
| 6.2.3 | <i>Etapa Constructiva</i> | 48 |
| 6.2.4 | <i>Etapa de Búsqueda Local</i> | 48 |
| 6.3. | ANÁLISIS DEL PEOR DE LOS CASOS | 50 |
| 6.4. | EXPERIMENTOS..... | 50 |
| 6.4.1 | <i>Implementación</i> | 50 |
| 6.4.2 | <i>Parametrización</i> | 51 |
| 6.4.3 | <i>Parámetro α</i> | 51 |
| 6.4.4 | <i>Parámetro Max_iter</i> | 52 |
| 6.5. | GENERACIÓN DE INSTANCIAS | 53 |
| 6.5.1 | <i>Instancias de comparación con el modelo de programación lineal entero</i> | 54 |
| 6.5.2 | <i>Instancias de mayor tamaño para el algoritmo GRASP</i> | 54 |
| 6.6. | MEDIDAS DE DESEMPEÑO..... | 55 |
| 6.6.1 | <i>Pacientes no programados por periodo de programación</i> | 55 |
| 6.6.2 | <i>Utilización de salas quirúrgicas</i> | 55 |
| CAPÍTULO 7. RESULTADOS Y ANÁLISIS | | |
| 7.1. | INTRODUCCIÓN..... | 56 |
| 7.2. | COMPARACIÓN CON EL MODELO DE PROGRAMACIÓN LINEAL ENTERO EN INSTANCIAS AJUSTADAS A LA REALIDAD | 56 |
| 7.3. | COMPARACIÓN CON EL MODELO IP EN INSTANCIA CON DATOS REALES..... | 58 |
| 7.3.1 | <i>Tablas operatorias</i> | 59 |

| | | |
|---|---|----|
| 7.3.2 | <i>Utilización de las salas quirúrgicas</i> | 62 |
| 7.4. | INSTANCIAS DE MAYOR TAMAÑO | 63 |
| 7.4.1 | <i>Tablas operatorias instancia IR100_15M</i> | 64 |
| 7.4.2 | <i>Tablas operatorias instancia IR150_15M</i> | 69 |
| 7.4.3 | <i>Utilización de salas quirúrgicas para instancias IR100_15M a IR250_22M</i> | 74 |
| CAPÍTULO 8. CONCLUSIONES Y TRABAJO FUTURO | | 76 |
| BIBLIOGRAFÍA | | 78 |
| ANEXO A. INSTANCIAS | | 83 |
| A.1. | INSTANCIA PCE 4 | 83 |
| A.2. | INSTANCIA PCE 17 | 85 |
| A.3. | INSTANCIAS REALES DE MAYOR TAMAÑO | 87 |



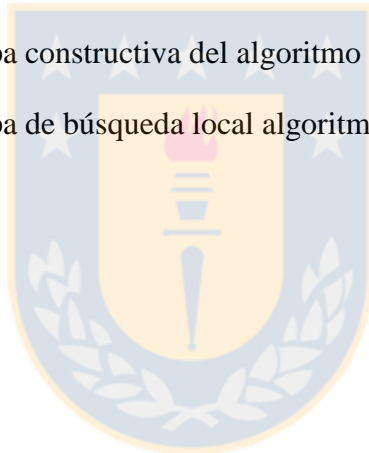
Lista de Tablas

| | |
|--|----|
| TABLA 4.1. LEYENDA TABLA 4.2. | 35 |
| TABLA 4.2. TIPOS DE RESTRICCIONES DEL PROBLEMA CONSIDERADAS EN LA LITERATURA | 35 |
| TABLA 4.3. MÉTODO DE RESOLUCIÓN UTILIZADOS EN LA LITERATURA..... | 37 |
| TABLA 5.1. ÍNDICES DEL MODELO MATEMÁTICO..... | 40 |
| TABLA 5.2. PARÁMETROS UTILIZADOS POR EL MODELO MATEMÁTICO | 40 |
| TABLA 5.3. VARIABLES DE DECISIÓN UTILIZADAS EN EL MODELO MATEMÁTICO | 41 |
| TABLA 6.1. RESULTADOS DE LA PARAMETRIZACIÓN DE α | 52 |
| TABLA 6.2. RESULTADOS DE LA PARAMETRIZACIÓN DE MAX_ITER | 53 |
| TABLA 7.1. CARACTERÍSTICAS DE LAS INSTANCIAS AJUSTADAS A LA REALIDAD | 57 |
| TABLA 7.2. RESULTADOS DE INSTANCIAS DE COMPARACIÓN..... | 58 |
| TABLA 7.3.RESULTADOS OBTENIDOS EN INSTANCIA DE COMPARACIÓN DE LOS MÉTODOS CON DATOS REALES | 59 |
| TABLA 7.4. TABLA OPERATORIA PARA LA SALA 1, EXCLUSIVA DE LA ESPECIALIDAD DE OTORRINOLARINGOLOGÍA | 60 |
| TABLA 7.5. TABLA OPERATORIA PARA LA SALA 2, EXCLUSIVA DE LA ESPECIALIDAD TRAUMATOLOGÍA..... | 61 |
| TABLA 7.6. TABLA OPERATORIA PARA LA SALA 3, CONSIDERADA COMO LIBRE..... | 62 |
| TABLA 7.7. UTILIZACIÓN DE LAS SALAS QUIRÚRGICAS INSTANCIA PCE17..... | 62 |
| TABLA 7.8. RESUMEN RESULTADOS OBTENIDOS EN INSTANCIAS IR100_15M HASTA IR250_22M | 64 |
| TABLA 7.9. TABLA OPERATORIA IR100_15M PARA EL DÍA LUNES | 65 |
| TABLA 7.10. TABLA OPERATORIA IR100_15M PARA EL DÍA MARTES..... | 66 |
| TABLA 7.11. TABLA OPERATORIA IR100_15M PARA EL DÍA MIÉRCOLES | 67 |
| TABLA 7.12. TABLA OPERATORIA IR100_15M PARA EL DÍA JUEVES | 68 |
| TABLA 7.13. TABLA OPERATORIA IR100_15M PARA EL DÍA VIERNES | 69 |

| | |
|--|----|
| TABLA 7.14. TABLA OPERATORIA IR150_15M PARA EL DÍA LUNES | 70 |
| TABLA 7.15. TABLA OPERATORIA IR150_15M PARA EL DÍA MARTES..... | 71 |
| TABLA 7.16. TABLA OPERATORIA IR150_15M PARA EL DÍA MIÉRCOLES | 72 |
| TABLA 7.17. TABLA OPERATORIA IR150_15M PARA EL DÍA JUEVES | 73 |
| TABLA 7.18. TABLA OPERATORIA IR150_15M PARA EL DÍA VIERNES | 74 |
| TABLA 7.19. UTILIZACIÓN SALAS QUIRÚRGICAS INSTANCIAS IR100_15M A IR250_22M..... | 75 |
| TABLA A.1. CARACTERÍSTICAS TAMAÑO INSTANCIA PCE 4..... | 83 |
| TABLA A.2 TABLA DE COMPATIBILIDAD, INSTANCIA PCE4..... | 83 |
| TABLA A.3. TABLA DE DISPONIBILIDAD HORARIA PARA EL DIA 1, INSTANCIA PCE 4..... | 84 |
| TABLA A.4. TABLA DE DISPONIBILIDAD HORARIA PARA EL DÍA 2, INSTANCIA PCE4 | 84 |
| TABLA A.5. COSTO UTILIZACIÓN DE SALAS QUIRÚRGICAS EN SOBRETIEMPO, INSTANCIA PCE4 | 84 |
| TABLA A.6. CARACTERÍSTICAS TAMAÑO DE LA INSTANCIA PCE 17 | 85 |
| TABLA A.7. TABLA DE COMPATIBILIDAD INSTANCIA PCE 17..... | 85 |
| TABLA A.8 TABLA DE DISPONIBILIDAD HORARIA PARA EL DIA 1, INSTANCIA PCE17 | 86 |
| TABLA A.9. TABLA DE DISPONIBILIDAD HORARIA PARA EL DIA 2, INSTANCIA PCE17..... | 86 |
| TABLA A.10. COSTO UTILIZACIÓN DE SALAS QUIRÚRGICAS EN SOBRETIEMPO, INSTANCIA PCE17..... | 87 |
| TABLA A.11.CARACTERÍSTICAS TAMAÑO DE LA INSTANCIA IR250_22M | 87 |
| TABLA A.12. COSTO UTILIZACIÓN DE SALAS QUIRÚRGICAS EN SOBRETIEMPO, INSTANCIA IR250_22M..... | 88 |
| TABLA A.13. TABLA DE COMPATIBILIDAD INSTANCIA IR250_22M..... | 89 |

Lista de Figuras

| | |
|--|----|
| Figura 1. Diagrama metodología de trabajo | 5 |
| Figura 2. Clasificación de pacientes | 7 |
| Figura 3. Proceso de planificación de cirugías | 10 |
| Figura 4. Red para el problema de asignación..... | 17 |
| Figura 5. Pseudocódigo algoritmo grasp | 45 |
| Figura 6. Estructura de los elementos en el algoritmo grasp | 47 |
| Figura 7. Ejemplo de una representación de un elemento en el algoritmo grasp | 47 |
| Figura 8. Pseudocódigo etapa constructiva del algoritmo grasp | 49 |
| Figura 9. Pseudocódigo etapa de búsqueda local algoritmo grasp | 49 |



Abreviaciones

Mayúsculas

| | |
|------------|---|
| N.A.D.W. | : need adjusted waiting days. |
| G.L.P.K | : GNU linear programming kit. |
| G.E.S. | : garantías explícitas en salud. |
| F.I.F.O. | : first in first out. |
| B&B | : branch and bound. |
| M.O.O. | : multi-objective optimization. |
| F.H.A. | : Fraser Health Authority. |
| G.A.M.S. | : general algebraic modeling system. |
| C.G.B.H. | : column generation based heuristic. |
| H.G.A. | : hybrid genetic algorithm. |
| R.C.P.S.P. | : resource constrained project scheduling. |
| M.S.S. | : master surgical scheduling. |
| M.I.P. | : mixed integer programming. |
| F.F.D. | : first fit decreasing. |
| L.P. | : linear programming. |
| P.P. | : prueba ponderadores. |
| P.C.E. | : programación de pacientes electivos. |
| G.R.A.S.P | : greedy randomized adaptive search procedures. |



Capítulo 1. Introducción

1.1. Introducción general

Los hospitales públicos y privados en todo el mundo han tenido un gran desafío en los últimos años para entregar un buen de servicio con recursos limitados, debido a los grandes avances en tecnología para la medicina, infraestructura hospitalaria y el aumento sostenible en la demanda. Adicionalmente, la existencia de listas de espera para el acceso a las intervenciones quirúrgicas como otros servicios, son un problema frecuente e importante, en particular, en los sistemas de salud públicos en Chile. En relación a esto, la Investigación de Operaciones entrega herramientas para la mejora de programación de procesos o actividades con el fin de usar de manera eficiente los recursos.

La unidad de pabellón, la cual consiste en salas quirúrgicas y salas de recuperación, es uno de los recursos más importantes y escasos dentro de cualquier hospital (Gordon, Paul, Lyles, & Fountain, 1988), posee entre el 60% y 70% de las admisiones a un hospital y sus costos son sobre el 40% del presupuesto anual (Denton, Viapiano, & Vogl, 2007). Las salas quirúrgicas, se definen como recintos especialmente diseñados y equipados para realizar actividades anestésicas y/o quirúrgicas.

En la programación de cirugías se pueden encontrar dos tipos de pacientes. Los pacientes electivos, son aquellos que se someterán a algún procedimiento o intervención quirúrgica y se realiza de forma planificada. Los pacientes de urgencia, son aquellos que ameritan una atención inmediata, por lo que poseen una alta prioridad en la programación de las salas quirúrgicas. Esta prioridad de pacientes, afecta tanto a los pacientes electivos como de emergencia y recae en factores tales como: la gravedad de la enfermedad, el efecto sobre la funcionalidad del cuerpo humano, factores sociales en los que está sometido el paciente, entre otros.

La programación de cirugías electivas, es un proceso complejo, especialmente porque considera una variedad de especialidades quirúrgicas, prioridades por el servicio y capacidad post-quirúrgica (Santibáñez, Begen, & Atkins, 2007). Además, existen muchas tareas complejas, por sus diversas partes interesadas (pacientes, administradores de salas quirúrgicas,

enfermeras, cirujanos, anestesistas, etc.) (Guerriero & Guido, 2011). La programación óptima de cirugías electivas se ve influida por diferentes factores, por ejemplo: el personal médico (cirujanos, anestesistas, enfermeras, arsenaleras, etc.); instalaciones (equipo especializado, capacidad pre-quirúrgica, etc.); múltiples salas quirúrgicas; capacidad de la sala de recuperación, cuidados intensivos y camas de hospitalización (Guerriero & Guido, 2011).

El principal objetivo de los directores de hospital es asegurar un mejor uso de los recursos médicos, realizar las intervenciones quirúrgicas a tiempo, maximizar el número de pacientes atendidos sin incurrir en gastos adicionales como por ejemplo el pago de sobretiempo al personal. Por lo tanto, la importancia de la programación de cirugías radica por un lado, en la satisfacción al paciente, el cual debe ser tratado prontamente. Y desde el punto de vista de la organización, una buena programación de cirugías y asignación de salas quirúrgicas permite lograr metas de desempeño, aminorar costos y mejorar la imagen de la organización.

1.2. Hipótesis de trabajo

La hipótesis de trabajo de esta investigación consiste en verificar si es posible, mediante un algoritmo GRASP, resolver el problema de programación de cirugías electivas y asignación de médicos y salas quirúrgicas para un hospital público chileno.

1.3. Alcances y limitaciones

En este trabajo se aborda el problema de programación de cirugías electivas de pacientes de hospitalización, asignación de salas quirúrgicas y priorización de pacientes electivos realizando un estudio del estado del arte, considerando ciertos puntos de interés para este estudio, profundizado sólo en el área de métodos determinísticos en investigación de operaciones.

El problema de programación de cirugías electivas es considerado como *NP-Hard* según (Vijayakumar, Parikh, Scott, Barnes, & Gallimore, 2013) donde el modelo de programación entero mixto es análogo al problema *dual bin-packing*. El modelo de programación lineal presentado es implementado y resuelto en instancias pequeñas (escenarios con baja cantidad de

variables y/o restricciones), debido al desempeño del *solver* utilizado, y los tiempos de procesamiento excesivamente altos para instancias reales.

Debido a la complejidad de acceder a datos reales para este tipo de problema, ya sea por razones de resguardo de privacidad de la organización o porque no poseen la información necesaria para este trabajo, algunos parámetros son ajustados a los datos reales o a lo presentado por la literatura.

1.4. Objetivos

1.4.1 Objetivo general

Generar y resolver un algoritmo metaheurístico GRASP para el problema de programación de cirugías electivas en un hospital público chileno.

1.4.2 Objetivos específicos

- Realizar un estudio del estado del arte en el problema de programación de cirugías electivas en el área de optimización combinatoria y la metaheurística GRASP.
- Generar un modelo matemático capaz de resolver el problema de programación de cirugías electivas.
- Definir el (los) objetivo(s) del modelo matemático y sus restricciones.
- Generar y resolver instancias pequeñas para el modelo matemático del problema de programación de cirugías electivas.
- Realizar un algoritmo basado en GRASP el cual resuelva el problema en cuestión.
- Realizar una parametrización del algoritmo GRASP generado para la resolución del problema de programación de cirugías electivas.
- Comparar los resultados obtenidos entre el modelo matemático de optimización y el algoritmo GRASP con medidas de desempeño definidas.
- Generar y resolver instancias basadas en parámetros reales del problema de programación de cirugías electivas con el algoritmo GRASP.

- Analizar el desempeño del algoritmo GRASP para el problema de programación de cirugías electivas en instancias basadas en parámetros reales.

1.5. Metodología

El trabajo presentado en este informe es realizado generando la descripción del problema (Capítulo 2) de programación de cirugías electivas a partir de una revisión bibliográfica, donde se obtuvo gran parte de la información necesaria para el problema. Por otro lado, la descripción del problema se complementa y valida mediante visitas a la Unidad de Pabellón de un Hospital Público y un cuestionario realizado sobre la programación de cirugías. Dentro de la revisión bibliográfica se encuentran estudios realizados tanto en la región, país, como el resto del mundo.

Con el fin de realizar un acercamiento al enfoque utilizado en este trabajo y facilitar al lector la comprensión del estudio, se exhibe como marco teórico: el *Scheduling*, los Problemas de Asignación, la Optimización Multiobjetivo, Métodos de solución exactos y Metaheurísticas (Capítulo 3).

Se realiza un estudio del estado del arte (Capítulo 4), en base a los estudios considerados de mayor relevancia para el problema, de esta manera, conocer la actualidad del tratamiento del problema en el área de estudio.

El modelo matemático propuesto, es definido en un trabajo anterior (Cartes, 2014), para cuál fue necesario definir los objetivos y los conjuntos de restricciones que representen de manera completa el problema de programación de cirugías, además de los supuestos y consideraciones particulares al problema. El modelo de programación lineal entera es implementado en el *solver* GLPK (GNU *Linear Programming Kit*) (Capítulo 5).

Con respecto al algoritmo propuesto basado en la metaheurística GRASP, es necesario definir el algoritmo y sus distintas etapas. El algoritmo se constituye de dos grandes etapas descritas en el informe (Capítulo 6): construcción y búsqueda local. Además se muestra y describe el proceso de parametrización del algoritmo y experimentos realizados para conocer el desempeño de éste.

Luego, se analiza el desempeño del algoritmo en comparación a resultados obtenidos anteriormente con el modelo de programación lineal y frente a instancias de mayor tamaño utilizando datos reales de un hospital público chileno. Cada uno de los resultados entregados por las diferentes instancias fueron analizados, utilizando medidas de desempeño para la programación de cirugías electivas: cantidad de pacientes no programados y la utilización de las salas quirúrgicas. En el análisis de desempeño, se realiza una comparación del modelo matemático con respecto a los resultados obtenidos con la metaheurística propuesta bajo las mismas instancias y posteriormente los resultados bajo las instancias de mayor tamaño utilizando parámetros reales del problema de programación de cirugías electivas con el algoritmo GRASP (Capítulo 7).

Las conclusiones obtenidas con este estudio y el trabajo a futuro para el problema tratado (Capítulo 8).

Finalmente, para mayor comprensión de la metodología utilizada en este trabajo se presenta la Figura 1 a continuación.

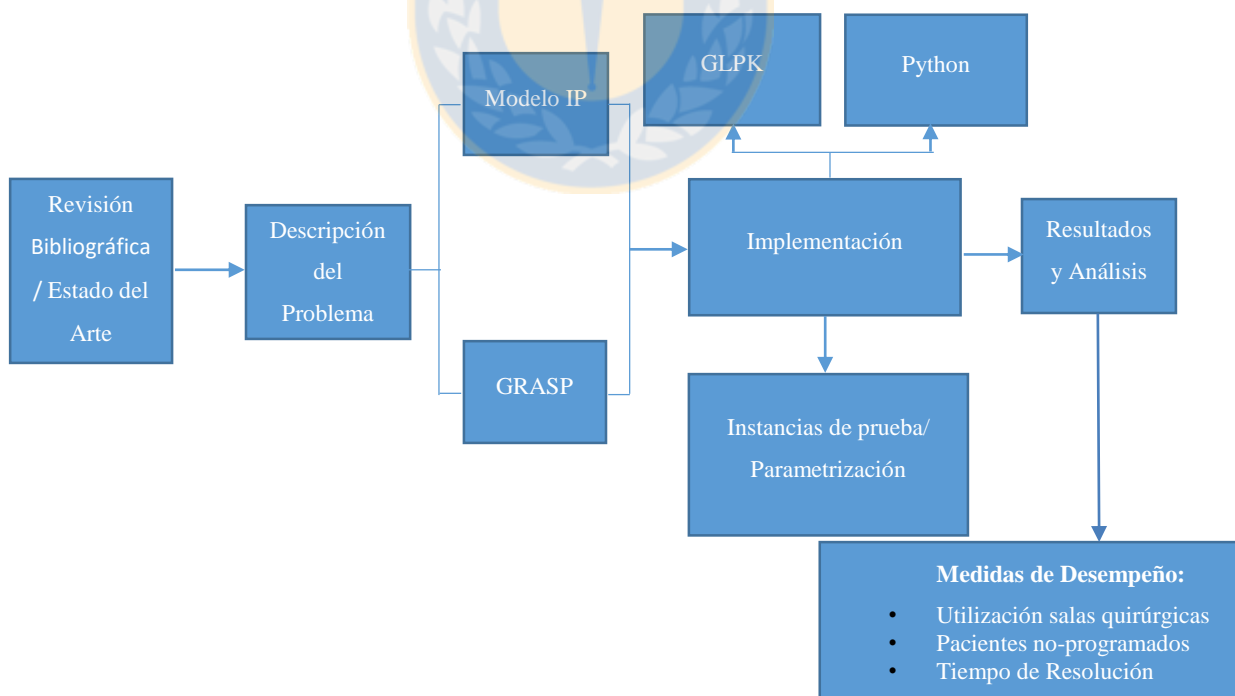


FIGURA 1. DIAGRAMA METODOLOGÍA DE TRABAJO

Capítulo 2. Descripción del problema

2.1. Introducción

El problema de programación de cirugías electivas consiste, en secuenciar las cirugías o casos quirúrgicos a realizarse durante un periodo de programación, a éstas cirugías se les debe asignar un médico principal que debe llevar a cabo dicha cirugía en conjunto con un equipo médico y una sala quirúrgica que cumpla los requerimientos para tal caso quirúrgico. La programación de cirugías electivas involucra diversos actores que participan en un mismo sistema. Cada uno de los actores otorga diferentes características que deben considerarse a la hora de realizar la programación de cirugías electivas. En este capítulo se describe cada participante a ser considerado en la programación de cirugías, en conjunto con la descripción y distinción de planificación y programación de cirugías y métodos o técnicas para la priorización de pacientes.

La descripción del problema expuesta a continuación, es basada en un trabajo realizado anteriormente (Cartes, 2014) para efectos de comparación de resultados mostrados más adelante en este informe.

2.2. Características de los pacientes

Existen dos grandes grupos de pacientes: los pacientes electivos, son aquellos que tienen cirugías planificadas con anterioridad; y los pacientes no-electivos que necesitan cirugía de manera inesperada y urgente (véase Figura 2).

Dentro de los pacientes electivos, se encuentran los pacientes de hospitalización (*inpatients*), que son aquellos que necesitan cuidados del hospital e inmediatamente son admitidos en la unidad de especialización correspondiente. Por otro lado, los pacientes ambulatorios (*outpatients*), los cuales después de ser intervenidos quirúrgicamente pueden retirarse a sus hogares.

En los pacientes no-electivos, se encuentran los pacientes urgentes y emergentes, estos se distinguen en la respuesta a la llegada del paciente al servicio (tiempo de espera hasta que

comienza la cirugía). El primer grupo debe atenderse lo antes posible y al segundo se le considera una estabilidad relativa en la cual se puede posponer la cirugía por un periodo de tiempo corto (Cardoen, Demeulemeester, & Beliën, 2010).

Cada paciente tiene asociado un tipo de cirugía a la cual será sometido, esta cirugía tiene características de compatibilidad con médicos y salas, es decir, cualquier médico no puede intervenir quirúrgicamente a los pacientes y cualquier sala quirúrgica no tiene las condiciones necesarias para cada cirugía.

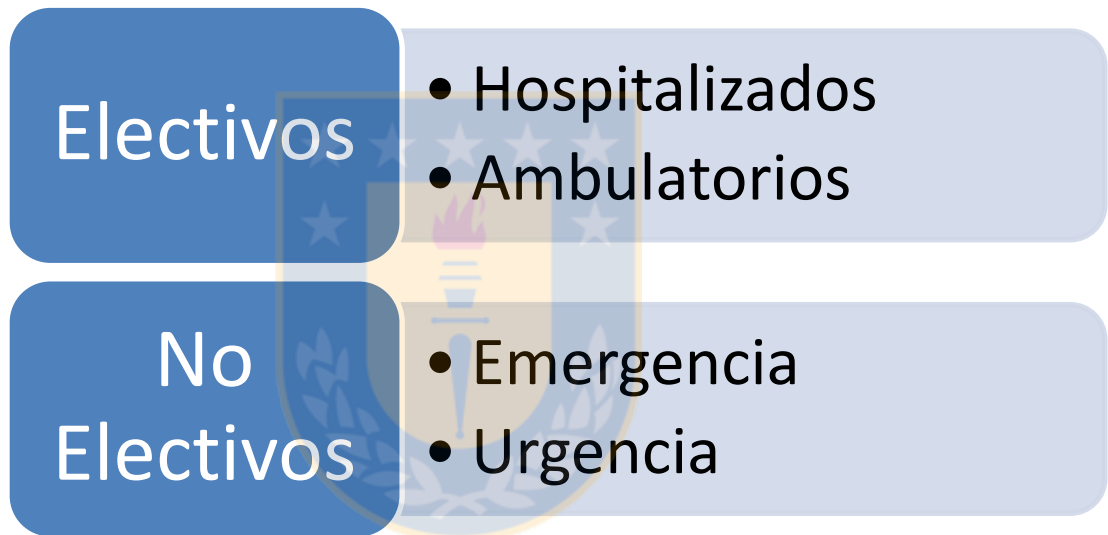


FIGURA 2. CLASIFICACIÓN DE PACIENTES

En particular, en el sector público chileno existen cirugías con un periodo de tiempo máximo en el que ésta debe ser atendida. Éste tipo de cirugías, son las cubiertas por las Garantías Explícitas en Salud (GES).

2.3. Características de la unidad de pabellón

La unidad de pabellón es la encargada de programar las cirugías. Esta unidad está en constante comunicación con servicios relacionados y servicios de apoyo pertenecientes a la misma organización.

La unidad de pabellón, está a cargo de la gestión de personal capacitado y asignación de éstos a las cirugías, donde se encuentran: médicos especialistas, médicos cirujanos, enfermeras, pabelloneras, anestesiastas, asistentes de anestesia y técnicos paramédicos. Además, de una gran variedad de recursos para llevar a cabo cada cirugía, tales como: recursos tecnológicos, insumos básicos, insumos estériles, entre otros.

2.4. Características de los médicos

Los médicos (cirujanos y especialistas), poseen cierta disponibilidad de horarios. Esta disponibilidad está presente en bloques de tiempo durante cada día de la semana. Generalmente, los médicos no atienden en un hospital público o privado durante todo el día.

Por otro lado, los médicos poseen atributos que les permiten o no realizar cirugías. Cada médico tiene asociado una especialidad, con competencias para atender a ciertos tipos de cirugía, es decir, éste actor del sistema posee compatibilidad con cada paciente.

Algunos médicos gestionan además, una demanda propia de intervenciones quirúrgicas. Aquí está la existencia de pacientes, en donde sus cirugías son programadas para un médico en particular independiente de que otro(s) médico(s) tenga(n) los atributos necesarios para atender a dicho paciente.

2.5. Características de las salas quirúrgicas

Las salas quirúrgicas también, poseen horarios disponibles. Algunas veces están disponibles, en media jornada, jornada completa o en bloques de tiempo. Poseen costos de utilización y características o condiciones que las hacen aptas o no para llevar a cabo las cirugías, es decir, también poseen compatibilidad con las cirugías.

Dentro de las principales características, se encuentra el equipamiento de las salas quirúrgicas, este equipamiento discrimina los tipos de cirugías que pueden ser realizados en cada sala quirúrgica (por ejemplo: cirugías laparoscópicas).

Para cada cirugía, es necesario considerar un tiempo de preparación y un tiempo de limpieza. Algunos tipos de cirugías pueden implicar tiempos más largos de preparación y

limpieza que otras, dado por: la complejidad, equipamiento médico, nivel de contaminación, entre otros. Los tiempos de preparación y limpieza prolongados pueden llevar a programar cirugías al principio o al final de la jornada laboral.

2.6. Características del equipo médico

El equipo médico o *staff* médico, está compuesto por: enfermera, pabellonera, arsenalera, anestesista, asistente de anestesia y técnicos paramédicos. Éstos deben estar presentes en cada cirugía, al menos al comienzo. En algunos casos, se permite realizar “cirugías paralelas”, siempre y cuando existan cirugías que permitan la presencia parcial de los miembros del equipo médico y los pabellones estén en una vecindad determinada. Además, se debe considerar que para cirugías determinadas cualquier miembro del equipo médico no las puede realizar, es decir, para algunos tipos de cirugías se requiere personal especializado o capacitado.

2.7. Planificación y programación de cirugías

En el proceso de programación de cirugías para pacientes electivos, se incluyen actividades que van desde determinar el tiempo en que los pabellones estarán operativos hasta la programación de casos individuales. Usualmente, este proceso consta de tres etapas (véase Figura 3) (Santibáñez et al., 2007)

1. Primero, se definen y asignan los tiempos de salas quirúrgicas a las diferentes especialidades (*Mix planning*).
2. La segunda etapa, especifica que especialidad usará cada sala quirúrgica cada día (*Block Schedule*).
3. La tercera etapa, que es donde se enmarca esta investigación es el *Patient Mix o Case Schedule* la cual programa casos individuales diariamente.

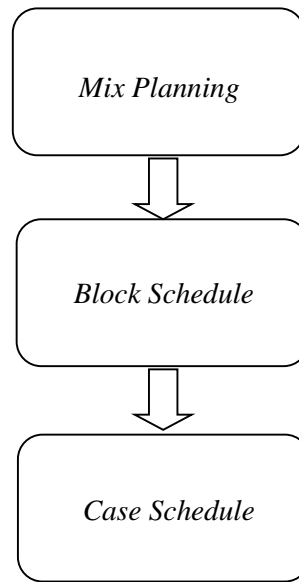


FIGURA 3. PROCESO DE PLANIFICACIÓN DE CIRUGÍAS

2.8. Modelos y estrategias de programación

En la actualidad, la programación de cirugías electivas es realizada de diferentes maneras, aunque generalmente se ejecuta según la experiencia del personal a cargo. Sin embargo, se puede encontrar en la literatura ciertos modelos y estrategias utilizadas para el problema de programación de cirugías electivas.

Se dispone de tres diferentes modelos clásicos de programación de cirugías electivas (Kharraja, 2003):

- *Programación Abierta*: Propone una programación en blanco en cada periodo de programación. Esta se completa por orden cronológico de acuerdo a los primeros en llegar.
- *Programación por Bloques*: Propone una programación estructurada, de acuerdo a una ubicación priorizada. Es decir, reserva ciertos bloques de tiempo para ciertos cirujanos. Generalmente se destinan media jornada o jornada completa.

- *Programación por Bloques Modificada*: Es una programación por bloques pero, existen bloques modificables en horas sin uso para otras cirugías. Además, es posible dejar horas libres para eventuales cirugías.

Cada uno de estos modelos puede ser combinado, con una de las tres siguientes estrategias de programación (Trilling, 2006):

- *Horas fijas*: Esta estrategia se centra en la utilización de los recursos, ya que no permite el uso de sobretiempo.
- *Cualquier día de trabajo*: Se centra en la satisfacción del paciente y del cirujano. El cirujano elige que día se realizará la intervención quirúrgica y se espera respetar esa fecha.
- *Tiempo razonable*: Esta estrategia consiste, en un acuerdo entre el médico y el hospital garantizando un determinado tiempo desde que el paciente espera hasta que es intervenido. No se involucra al cirujano en la elección de la fecha de la cirugía.

2.9. Priorización de pacientes electivos

En la literatura especializada, se encuentran diferentes enfoques y métodos para la priorización de pacientes electivos.

En algunos casos, los encargados de realizar la priorización de pacientes son los servicios de donde provienen estos pacientes, es decir, llegan ya priorizados a la unidad de pabellón donde sólo realizan la programación de cirugías (sin priorizar pacientes).

Capítulo 3. Marco Teórico

3.1. Introducción

En esta sección, se presentan las principales temáticas que permiten abordar el problema de programación y asignación de cirugías electivas desde el enfoque utilizado en el presente trabajo. El problema tratado se identifica como una combinación de dos grandes áreas de la optimización combinatoria, en primer lugar se presenta el área de *scheduling* (Sección 3.2) el cual permite programar tareas a lo largo del tiempo, es decir, decidir qué cirugías serán realizadas y en qué momento. Por otro lado, los problemas de asignación (Sección 3.3), que permiten para este trabajo obtener la mejor asignación de médicos y salas quirúrgicas dadas las condiciones propias del problema.

Ya que el problema se modela como un problema de programación multiobjetivo (Sección 3.4), se presenta además, los principales métodos para abordar dichos problemas.

Además, se presentan en éste capítulo los métodos metaheurísticos, su clasificación (Sección 3.5) y la metaheurística *Greedy Randomized Adaptive Search Procedures* (GRASP), el cual es utilizado para abordar el problema de programación de cirugías electivas y asignación de salas quirúrgicas de manera metaheurística (Sección 3.6).

Finalmente, en la Sección 3.7 se describen los métodos de resolución, clasificados en tres tipos: Heurísticas, algoritmos aproximados y métodos exactos.

3.2. Scheduling (Pinedo, 2008)

El *scheduling* es el proceso en que se asignan recursos escasos a tareas a lo largo del tiempo.

Los recursos pueden tomar diferentes formas tales como: máquinas en un taller, pistas de aterrizaje en un aeropuerto, turnos de personal en un hospital, etc. En investigación de operaciones, el *scheduling* es un área ampliamente estudiada en las últimas décadas y tiene como finalidad optimizar uno o más objetivos.

En todo problema de *scheduling* se considera un número de trabajos y máquinas finito. El número de trabajos se denota como n y el número de máquinas como m . Generalmente el subíndice j se refiere a trabajos y/o tareas, y el subíndice i a una máquina.

Los siguientes parámetros, son asociados a un trabajo j :

- **Tiempo de proceso (p_{ij}):** representa el tiempo requerido para procesar un trabajo j en una máquina i .
- **Fecha de liberación (r_j):** se refiere al momento en que un trabajo j está preparado para ser procesado, es decir, el momento de arribo al sistema.
- **Fecha de entrega (d_j):** representa el compromiso entre el productor y el cliente, es decir, la fecha en que el trabajo estará completo y puede ser enviado o entregado al cliente. El trabajo puede ser completado luego de su fecha de entrega, pero existe una penalización por ello.
- **Prioridad o ponderación (w_j):** La prioridad de un trabajo j representa la importancia relativa de éste sobre otros trabajos en el sistema.

Un problema de *scheduling* es descrito por un triplete $\alpha | \beta | \gamma$, notación introducida por (Graham, Lawler, Lenstra, & Kan, 1979). El campo α representa la configuración del sistema de trabajo o tipo de proceso. El campo β , señala las condiciones y características del procesamiento. Y el campo γ describe el objetivo a optimizar.

Los posibles casos que pueden ser especificados en el campo α son:

- **Máquina única (I):** Es el caso más simple y es un caso especial en comparación con otros más complicados.
- **Máquinas paralelas idénticas (P_m):** Existen m máquinas paralelas idénticas. El trabajo j requiere una única operación y puede ser procesado en cualquier máquina del sistema, o cualquier máquina que pertenezca a un subconjunto dado.
- **Máquinas paralelas con distintas velocidades (Q_m):** Existen m máquinas paralelas con diferentes velocidades, independiente de los trabajos. La velocidad de la máquina i es denotada como v_i . Este caso es referido como máquinas uniformes.

- **Máquinas paralelas no relacionadas (R_m):** Este caso representa cuando la velocidad de las máquinas depende del trabajo que se está procesando.
- **Flow shop (F_m):** Existen m máquinas en serie y cada trabajo debe ser procesado en cada máquina. Todos los trabajos deben realizar la misma ruta, luego de terminar de procesar un trabajo en una máquina pasa a la espera en la próxima máquina.
- **Flexible flow shop (FF_c):** Es una generalización del caso del *flow shop* y las máquinas paralelas. Existen distintas etapas con m máquinas paralelas idénticas y cada trabajo debe seguir en serie por estas etapas.
- **Job shop (J_m):** Cada trabajo tiene su propia ruta. En el *job shop* se utilizan estaciones de trabajo. Existe una distinción entre *job shops* en los cuales cada trabajo debe estar necesariamente en todas las máquinas y *job shops* en los cuales no es necesario que cada trabajo visite cada máquina.
- **Flexible job shop (FJ_c):** Es una generalización del caso de *job shop* y máquinas paralelas. Existen diferentes etapas compuestas por m máquinas en donde cada trabajo tiene su propia ruta.
- **Open shop (O_m):** Cada trabajo debe ser procesado en cada máquina. Sin embargo, algunos de estos tiempos de procesamiento pueden ser cero. No existen restricciones con respecto a la ruta de cada trabajo a través de las máquinas del sistema.

Las restricciones de procesamiento son especificadas en el campo β , donde pueden haber múltiples entradas. En este campo se puede especificar:

- **Fechas de liberación (r_j):** Si este símbolo está presente en el campo β , el trabajo j no puede realizarse antes de ésta fecha. Si no está la restricción de fechas de liberación, entonces el trabajo puede procesarse en cualquier momento.
- **Secuencia dependiente del tiempo de preparación (s_{jk}):** s_{jk} representa la secuencia dependiente del tiempo de *setup* entre los trabajos j y k . Si esta restricción no está incluida en el campo correspondiente, los tiempos de *setup* para cada trabajo son iguales a cero.

- **Preemptions (*prmp*):** Está permitido interrumpir el procesamiento de un trabajo en cualquier momento y colocar otro trabajo en la máquina en cuestión.
- **Restricciones de precedencia (*prec*):** Se requiere que uno o más trabajos sean realizados para que otro trabajo pueda comenzar a procesarse.
- **Breakdowns (*brkdown*):** Implica que las máquinas no están disponibles continuamente.
- **Restricciones de elección de máquina (M_j):** Este símbolo puede aparecer cuando se trata de casos con máquinas paralelas. No todas las máquinas son capaces de procesar un trabajo j .
- **Permutación (*prmu*):** Es una restricción que puede aparecer en un caso de *flow shop*, donde las colas de espera en cada máquinas actúan de acuerdo a la regla *Primero en Entrar Primero en Salir* (FIFO, por sus siglas en inglés, *First In First Out*).
- **No-wait (*nwt*):** Este símbolo puede aparecer en casos de *flow shops*, donde a los trabajos no se les permite esperar entre dos máquinas sucesivas.
- **Recirculación (*recrc*):** Puede ocurrir en un *job shop* o un *job shop* flexible, cuando un trabajo puede visitar una máquina o estación de trabajo más de una vez.

En el tercer campo (γ), se ubican el (los) objetivo(s) a optimizar, ya sea maximizar o minimizar. Algunos de los principales objetivos a minimizar en problemas de *scheduling* son según (Pinedo, 2008):

- **Makespan (C_{max}):** Se define como el tiempo en que el último trabajo deja el sistema, es decir, el tiempo en que se terminan de procesar todos los trabajos. Una minimización del *makespan* usualmente implica una alta utilización de las máquinas.
- **Tardanza Máxima (L_{max}):** Permite medir la peor violación de las fechas de entrega, donde la tardanza se define por la ecuación (3.1).
- **Total Weighted Completion Time ($\sum w_j C_j$):** La suma de las ponderaciones de tiempos de término de los n trabajos, otorga la tenencia total o costos de inventario incurridos en la programación. Cabe destacar que $\sum C_j$ se le conoce como *Tiempo de Flujo*.
- **Tardanza Total Ponderada ($\sum w_j T_j$):** Se le otorga a cada trabajo una ponderación, la cual implica la importancia de entregar cada trabajo antes de la fecha de entrega.

- **Número Ponderado de Trabajos Tardíos ($\sum w_j U_j$):** Cada trabajo que termine posterior a la fecha de entrega se considera como trabajo tardío, véase ecuación (3.2). Esta medida de desempeño es utilizada no sólo en el ámbito académico, sino que también es utilizada a menudo en la práctica (Pinedo, 2008).

$$L_j = C_j - d_j \quad (3.1)$$

$$U_j = \begin{cases} 1, & \text{si } C_j > d_j \\ 0, & \text{otro caso} \end{cases} \quad (3.2)$$

3.3. Problemas de asignación (Burkard, Dell’Amico, & Martello, 2009)

Los problemas de asignación son una de las tantas aplicaciones de la programación binaria, además de ser uno de los problemas más antiguos y altamente estudiados en la optimización combinatoria. Los problemas de asignación tratan de representar casos en donde n ítems deben asignarse a n objetos, con la finalidad (generalmente) de minimizar los costos de realizar dicha asignación.

El problema de asignación, nace del problema de transporte clásico, con la diferencia en que el número de destinos deben ser igual al número de fuentes. Los problemas de asignación tienen muchas aplicaciones tales como: programación y planificación de la producción, telecomunicaciones, economía, etc.

3.3.1 Problemas de asignaciones lineales

Sea c_{ij} el costo incurrido por la asignación del objeto i al objeto j , donde se busca minimizar los costos de asignación.

El problema de asignación puede ser representado como la red expuesta en la Figura 4

El problema consta en asignar a cada tarea una máquina que realice dicha tarea. Cada máquina posee un costo de asignación a cada operario, así uno de los objetivos a buscar en este problema puede ser minimizar el costo de realizar la asignación.

Las restricciones del problema, indican que cada máquina puede ser asignada solo a un operario, y un operario puede operar sólo una máquina. De esta manera el modelo matemático para este problema es el siguiente:

$$\text{Min } \sum_i \sum_j c_{ij} x_{ij} \quad (3.3)$$

sujeto a:

$$\sum_i^n x_{ij} = 1 \quad \forall j = 1, \dots, n \quad (3.4)$$

$$\sum_j^n x_{ij} = 1 \quad \forall i = 1, \dots, n \quad (3.5)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \quad (3.6)$$

Donde c_{ij} es el costo de asignar la máquina i al operario, y x_{ij} es la variable de decisión binaria que toma valor 1 cuando la máquina i se asigna al operario j , 0 en otro caso.

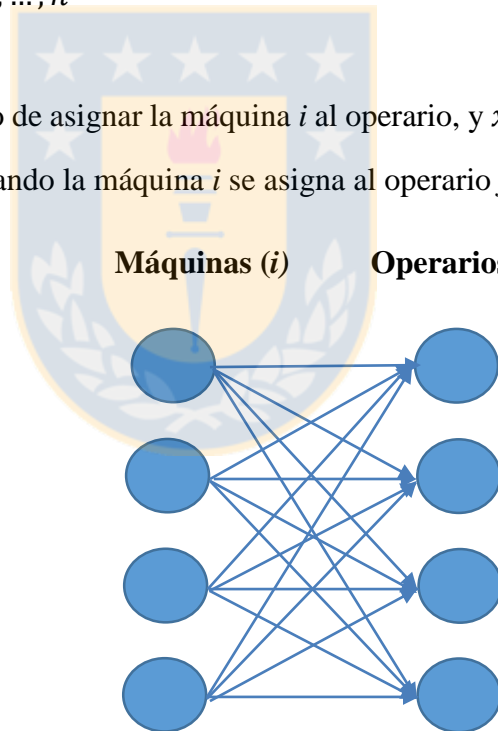


FIGURA 4. RED PARA EL PROBLEMA DE ASIGNACIÓN

Este problema posee una propiedad combinatoria de unimodularidad (el determinante de cualquier submatriz cuadrada tiene valor 0, +1 o -1) en la matriz A , de restricciones (Burkard et al., 2009), por lo cual la solución del relajamiento continua es entera y óptima. Sin embargo

los principales algoritmos para resolver este problema son los algoritmos primal y dual, y el método Húngaro (Burkard et al., 2009), con una complejidad de $O(n^3)$.

3.4. Optimización multiobjetivo (Marler & Arora, 2004)

El proceso de optimizar sistemáticamente y simultáneamente un conjunto de funciones objetivos es llamado Optimización Multiobjetivo (MOO, por las siglas en inglés de *Multiobjective Optimization*) u Optimización Vectorial.

La optimización multiobjetivo se define por (Osyczka, 1985) como el problema de encontrar un vector de variables de decisión que satisfagan un cierto conjunto de restricciones y optimicen un conjunto de funciones objetivo. Estas funciones forman una descripción matemática de los criterios de desempeño que suelen estar en conflicto unos con otros y que se suelen medir en unidades diferentes.

La optimización multiobjetivo originalmente nace de tres áreas: Teorías económicas de equilibrio y bienestar, teoría de juegos y matemática pura.

3.4.1 Definición del problema de optimización multiobjetivo

El problema general de optimización multiobjetivo se define de la siguiente manera (Marler & Arora, 2004):

$$\text{Min } F(x) = [F_1(x), F_2(x), \dots, F_k(x)]^T \quad (3.7)$$

$$\text{Sujeto a } g_j(x) \leq 0, \quad j = 1, 2, \dots, m \quad (3.8)$$

$$h_l(x) = 0, \quad l = 1, 2, \dots, e \quad (3.9)$$

Donde k es el número de funciones objetivos, m es el número de restricciones de desigualdades y e es el número de restricciones de igualdades. $x \in E^n$ es un vector de variables de decisión, donde n es el número de variables x_i . $F(x) \in E^n$ es un vector de funciones objetivo, estas son llamadas también: *criterios, funciones de costo, o funciones de valor*.

3.4.2 Optimalidad de Pareto

En contraste a la Optimización Mono-objetivo, una solución a un problema multiobjetivo es más que una definición, un concepto. Generalmente no existe una solución global, y en gran parte de los casos es necesario determinar un conjunto de puntos que encajen en una definición determinada de un óptimo (Marler & Arora, 2004)

La optimalidad de Pareto se define como: un punto $x^* \in X$, es un óptimo de Pareto si y solo si no existe otro punto $x \in X$, tal que $F(x) \leq F(x^*)$, y $F_i(x) \leq F_i(x^*)$ para al menos una función objetivo, donde X es el espacio de soluciones.

Los vectores correspondientes a las soluciones incluidas en el conjunto de óptimos de Pareto son llamados *no dominados*. La gráfica de funciones objetivo cuyos vectores no dominados se encuentran en el conjunto de óptimos de Pareto se le denomina *frente (o frontera) de Pareto*.

3.4.3 Métodos de optimización multiobjetivo

Entre los métodos de optimización multiobjetivo se pueden destacar tres categorías. Métodos con articulación de preferencias a priori, Métodos con articulación de preferencias a posteriori, Métodos sin articulación de preferencias.

Los métodos con articulación de preferencias a *priori*, permiten al tomador de decisiones especificar preferencias, que son articuladas con un nivel de importancia relativo con los diferentes objetivos. Los métodos con articulación de preferencias a *posteriori*, permiten al tomador de decisiones elegir la mejor solución desde un conjunto de soluciones generadas, debido a que en muchos casos es complejo para el tomador de decisiones entregar una aproximación explícita de las preferencias en términos de objetivos. Los métodos sin articulación de preferencias, son diseñados para los casos en donde el tomador de decisiones no puede definir concretamente lo que prefiere, estos métodos son generalmente simplificaciones de los métodos a *priori* (Marler & Arora, 2004), donde principalmente se excluyen los parámetros de dichos métodos.

En esta sección se abordarán 2 de los métodos más utilizados en la literatura, pero que no quiere decir que entreguen todos los puntos pertenecientes a la frontera de Pareto. Además se expone un tercer método que entrega todos los puntos extremos eficientes de un problema multiobjetivo.

- **Suma lineal de pesos:** Convierte el problema multiobjetivo en un problema mono-objetivo realizando una suma ponderada de los diferentes objetivos.

$$\sum_{i=1}^k \omega_i F_i(x) \quad (3.10)$$

Donde $\omega_i \geq 0$, son los pesos que representan la importancia relativa de cada uno de los k objetivos. La principal ventaja de este método es su simplicidad y eficiencia. La principal desventaja es la dificultad de definir un conjunto de ponderadores que permita generar una parte importante del frente de Pareto.

- **Restricción ε (ε -constraint):** Consiste en seleccionar una función objetivo de preferencia, y considerar el resto de las funciones objetivo como restricciones acotadas por niveles permisibles denominados ε_i . Posteriormente se alteran las cotas ε_i con el fin de generar un conjunto completo de Pareto.
- **Método simplex multiobjetivo:** Es el único método que garantiza la generación de todos los puntos extremos eficientes (Ehrgott, 2005). Consiste en encontrar todos los puntos de la frontera de Pareto del problema multiobjetivo desplazándose de un punto extremo a un punto extremo adyacente, donde el algoritmo del *método simplex* tradicional constituye el mecanismo adecuado para realizar dicho desplazamiento. Cabe mencionar que este método es muy costoso computacionalmente, y es utilizado solo para resolver problemas multiobjetivo muy pequeños.

3.5. Metaheurísticas (Gogna & Tayal, 2013)

En el mundo real, se encuentran una gran cantidad de problemas con interdependencia de variables, restricciones no lineales y con un espacio de solución muy grande. Las Metaheurísticas pertenecen a un grupo de métodos de resolución no exactos, las cuales otorgan un marco referencial para las heurísticas que pueden ser aplicadas a diversos problemas. Las

metaheurísticas, han tenido un gran desarrollo como soluciones a problemas del mundo real en las últimas dos décadas (Gogna & Tayal, 2013) y dentro de éstas metaheurísticas se encuentra la siguiente clasificación (Talbi, 2009):

- **Inspirados en la naturaleza versus no inspirados en la naturaleza**, por ejemplo: Algoritmos Evolutivos (GA) versus Búsqueda Local Iterativa (ILS).
- **Basados en población versus búsqueda en un punto**, por ejemplo: Algoritmos Genéticos (GA) versus Búsqueda Tabú (TS), *Simulated Annealing* (SA).
- **Iterativo versus codiciosos**, por ejemplo: *Simulated Annealing* (SA) versus Optimización de Colonia de Hormigas (ACO).
- **Dinámicos versus función objetivo estática**, por ejemplo: Búsqueda Local Guiada versus Optimización por Enjambre de Partículas (PSO).
- **Con uso de memoria versus sin uso de memoria**, por ejemplo: Búsqueda Tabú (TS), versus *Simulated Annealing* (SA).
- **Estructura de vecindarios única versus estructura de vecindarios múltiple**, por ejemplo: Búsqueda Local Iterativa (ILS) versus Búsqueda de Vecindarios Variable (VNS)

3.6. *Greedy Randomized Adaptive Search Procedures* (Resende & Ribeiro, 2003)

GRASP es una metaheurística multi-inicio para problemas combinatorios, en la cual, cada iteración consta de una fase de construcción y otra de búsqueda local. La fase de construcción crea soluciones factibles, cuyos vecindarios se investigan hasta encontrar un mínimo local en la fase de búsqueda local. La mejor solución de todas es la que se entrega como solución del algoritmo.

La fase de construcción, consiste en incorporar a la solución parcial un elemento del conjunto de elementos candidatos, los cuales deben ser factibles. Luego de incorporar un elemento a la solución parcial, los elementos restantes deben ser evaluados para no destruir la factibilidad. La evaluación del próximo elemento a elegir se realiza con una función de

evaluación *greedy*, la cual genera una lista reducida de candidatos (RCL, por sus siglas en inglés), que son los mejores elementos a incorporarse. Posteriormente, la fase de búsqueda local mejora la solución, ya que no es necesario que la etapa constructiva entregue el óptimo.

La fase de búsqueda local funciona de manera iterativa, donde se reemplaza la solución parcial por una solución mejor encontrada dentro del vecindario explorado.

Esta metaheurística posee diversas aplicaciones y estrategias para realizar mejoras en el desempeño (Festa & Resende, 2009). En (Park & Seo, 2012), resuelven el problema de programación del transporte y ruteo en un astillero, el cual es modelado como un problema de programación de máquinas paralelas con secuencia dependiente de la preparación y restricciones de precedencia. Con el objetivo de maximizar el balance de la carga laboral entre los transportadores, donde los resultados del algoritmo GRASP mantienen el *completion time* considerablemente bajo y un balance de la carga laboral de los transportadores para instancias reales, los cuales son simulados para realizar un análisis.

El trabajo de (Delorme, Gandibleux, & Rodriguez, 2004), se investiga el *set packing problem* en base a un algoritmo básico GRASP y diferentes maneras de mejoras. Estas mejoras consisten en un proceso de auto-parametrización (*reactive GRASP*), un aprendizaje del algoritmo mediante diversificación en la selección de elementos y un proceso de intensificación, a través de *path relinking*. Utilizando dos diferentes conjuntos de instancias resuelven el problema con el MIP- *solver* de CPLEX hasta el punto que fuera posible con un tiempo predeterminado y luego realizan las comparaciones con el algoritmo propuesto. A pesar de que en algunos casos, el algoritmo GRASP entrega resultados en tiempos computacionales altos, muestra ser más eficiente que CPLEX. Además proponen diferentes configuraciones del algoritmo GRASP que permitiría utilizar menos recursos (criterio de parada en el *path relinking*, y menor cantidad de iteraciones en el *local search*).

En el Capítulo 6 se profundizará en los componentes necesarios para la implementación del algoritmo GRASP para resolver el problema de programación de cirugías electivas con priorización de pacientes.

3.7. Métodos de resolución

Generalmente, se utilizan 3 tipos de métodos de resolución para resolver problemas NP-Hard (Ropke, 2005):

- **Heurísticas:** Son aquellos métodos de resolución que generalmente encuentran una solución factible rápidamente con una calidad de solución razonable. En estos métodos, no existe garantías acerca de la calidad de solución. Las heurísticas resuelven generalmente problemas reales debido a su velocidad y su capacidad de manejar grandes instancias. Un caso especial de heurísticas son las metaheurísticas, estas otorgan un marco referencial para heurísticas que pueden ser aplicadas a diversos problemas.
- **Algoritmos aproximados:** Son un caso especial de heurísticas, que otorgan una solución y error garantizado. Por ejemplo, un método puede garantizar que la solución obtenida es al menos k veces más costosa que la mejor solución asequible. Un ejemplo de un algoritmo aproximado es el algoritmo Greedy (Vazirani, 2001).
- **Métodos exactos:** Estos métodos garantizan que la solución óptima se encontrará si es que se le otorga el suficiente tiempo y espacio (memoria).

Capítulo 4. Estado del Arte

4.1. Introducción

En este capítulo se mostrará el estado del arte en la programación de cirugías usando Métodos Determinísticos en Investigación de Operaciones. Se presentan, las publicaciones locales e internacionales consideradas relevantes para este trabajo, y se analizan de acuerdo a los modelos matemáticos propuestos por los respectivos autores, con el fin de conocer los principales supuestos y restricciones del problema. Por otro lado, se destacan los métodos de solución del problema, para conocer el enfoque y direcciones recientes que ha tomado este tema de investigación.

Las publicaciones internacionales comprenden, desde el año 2007 en adelante y fue realizada con el fin de complementar la revisión bibliográfica realizada en el trabajo anterior (Cartes, 2014). La búsqueda es relacionada con el manejo de salas quirúrgicas, programación de cirugías electivas, planificación de cirugías electivas y *scheduling*.

4.2. Revisión Bibliográfica

4.2.1 Trabajos locales

En (Latorre, 2011), se secuencia un conjunto de cirugías, asignándolas a las salas quirúrgicas de un hospital y posterior una asignación de camas de recuperación. Se modela el problema como *flow shop* flexible de dos etapas, con máquinas paralelas no relacionadas en ambas etapas, tiempos de *setup* dependientes de la secuencia y de la máquina.

Se implementó el modelo de programación lineal entero mixto en el lenguaje AMPL y fue resuelto con CPLEX 12.2.0.0. Y se realizó una heurística constructiva para instancias grandes programadas en C++ y Dev-C++. Las pruebas del modelo y heurística fueron realizadas con datos históricos obtenidos del Hospital Clínico de la Universidad de Chile, para comparar desempeño con las programaciones realizadas anteriormente. Se realizó una comparación en los resultados obtenidos con el modelo matemático propuesto con la heurística constructiva, en

donde el modelo no encontró solución entera en el 36% de los 400 casos experimentados aleatoriamente. En cambio, la heurística constructiva, muestra soluciones factibles en menos de un segundo de tiempo de ejecución.

En (Pradenas, Vidal, Parada, & Melgarejo, 2012) tienen como propósito resolver la programación semanal de cirugías y la asignación de cirujanos en un hospital público. El estudio es dividido en dos partes: primero, las cirugías fueron programadas con un modelo de programación lineal entera multi-knapsack; luego, los cirujanos fueron asignados mediante un método basado en una heurística cronológica backtracking. En la implementación computacional se usó librerías ILOG en el lenguaje C++.

Los autores comparan los resultados obtenidos con la programación manual realizada en el Hospital Guillermo Grant Benavente y muestra una mejora en el 17,53% en la cantidad de cirugías realizadas en el periodo de programación.

En el artículo de (Pradenas Rojas & Matamala Vergara, 2012), se utiliza un modelo de optimización multiobjetivo y un algoritmo metaheurístico implementado computacionalmente que permite la programación semanal de intervenciones quirúrgicas. Se utiliza una instancia real con datos históricos del Hospital Guillermo Grant Benavente, donde el tiempo de ejecución del algoritmo, implementado en C++ fue de 7 minutos para 191 cirugías en lista de espera, tiempo considerablemente menor al tiempo en que se realiza la programación de forma manual.

El modelo matemático propuesto consta de 3 objetivos: maximizar la sumatoria de los factores de urgencias de las cirugías, para priorizar a los pacientes que requieren una pronta atención; maximizar la suma del tiempo de espera de una cirugía, priorizando a los pacientes que llevan mayor tiempo de espera y maximizar el número de cirugías en el periodo de programación.

Para resolver el modelo multiobjetivo se analizan diferentes enfoques mostrados en la literatura para problemas multiobjetivos con algoritmos genéticos, mostrando awGA como el mejor en cuanto al tiempo de procesamiento (7 minutos), mejor calidad de soluciones y mejor flexibilidad bajo contingencias existentes.

En (Wolff, Durán, & Rey, 2012), se desarrolla y comparan 4 modelos de optimización que permiten la programación de pabellones durante intervalo de tiempos específicos (semanalmente). En los modelos se consideraron (de acuerdo a los hospitales estudiados) el espacio físico, los insumos, anestesistas, el equipamiento y el equipo médico (enfermera, arsenalera y pabellones).

Los modelos planteados, presentan la priorización de pacientes como el objetivo principal, y dividen este en 3 subproblemas. El primero pretende solucionar qué día se opera a cada paciente. El segundo, corresponde a la asignación de médicos a las intervenciones de manera equitativa y el tercer subproblema soluciona en qué orden se debe operar dentro del día.

Se utiliza un modelo de programación lineal entera, una variante del modelo de programación lineal entera (modificaciones con respecto a la prioridad y se divide el problema en asignación y programación), un algoritmo basado en un modelo de factibilidad y un algoritmo de tipo *backtracking*. En donde el último presenta las mejores características para resolver el problema.

Los resultados entregados por el artículo en casos reales, muestran una mejora entre un 10% y 15% en la utilización de salas quirúrgicas en comparación con los métodos manuales. Los autores realizan el trabajo en el Hospital Luis Calvo Mackenna, Santiago de Chile, utilizando datos históricos de dicho hospital prueban los modelos y algoritmos con instancias de 50, 100 y 200 pacientes, donde el algoritmo *backtracking* resuelve el problema en 4, 4 y 7 segundos respectivamente para cada instancia.

Cabe recalcar, que estos autores mencionan trabajar en una aplicación para dicho hospital y su implementación.

En (Cartes, 2014), se desarrolla un modelo de programación lineal entero multiobjetivo para la programación de cirugías electivas el cual considera restricciones de disponibilidad de médicos, disponibilidad de salas quirúrgicas, disponibilidad del *staff* médico, compatibilidad de médicos y salas quirúrgicas, balance de la carga laboral, y la consideración de la disponibilidad de camas, entre otros. Con los el objetivo de maximizar la prioridad de pacientes programados y minimizar el uso de sobretiempo.

El modelo se resuelve con GLPK (*GNU Linear Programming Kit*) que posee un método de resolución de modelos de programación entero basado en el método *Branch & Cut*. Entregando resultados óptimos en tiempos razonables para instancias pequeñas y resultados de mayor tiempo de ejecución en una instancia real de un hospital público chileno (36 horas de ejecución) de: 2 días, 15 pacientes, 5 médicos, 3 salas quirúrgicas y 20 periodos de tiempo, alcanzándose una utilización máxima del 87.5% para una sala quirúrgica

4.2.2 Priorización de pacientes

(Testi, Tanfani, Valente, Ansaldo, & Torre, 2008), plantean el problema de la espera por cirugías electivas con una comparación de dos enfoques: prioridad y urgencia. Esta comparación es desempeñada en términos de eficiencia (cantidad total de necesidades clínicas satisfechas con las mismas salas quirúrgicas y el mismo *staff* médico) y equidad (cómo los pacientes son admitidos de acuerdo a sus necesidades). Además, se realiza un asesoramiento de tratamiento de urgencia apuntado a categorizar pacientes en grupos de urgencia (URGs), con un tiempo de espera máximo recomendado, y la implementación de un algoritmo de puntuación para priorizar los pacientes, el cual apunta a determinar la prioridad relativa de cada paciente en la lista de espera, el orden y admisión correspondiente.

Se realiza la comparación de estos enfoques mediante una puntuación, que es realizado por la medida NADW (*Need-Adjusted-Waiting-Days*).

(Mullen, 2003), se realiza un análisis crítico en relación a los sistemas de priorización para pacientes electivos y de urgencia. Se destaca la importancia de conocer los objetivos que se buscan en un sistema de priorización, según los autores, es necesario diferenciar entre prioridad de urgencia y prioridad de importancia.

- **Prioridad de urgencia:** Se relaciona con las implicaciones que trae un retraso del tratamiento, el cual incluye muerte y otras adversidades; deterioración; sufrimiento/dolor e inhabilidad de funcionamiento. A pesar de los numerosos efectos del retraso de intervenciones, es clave conocer que en muchos casos las cirugías tempranas no reducen la mortalidad.

- **Prioridad de importancia:** puede ser usada para determinar si un tratamiento es ofrecido o no. Percibe la importancia del tratamiento mejor que la urgencia. Los factores a considerar son: severidad de la enfermedad, capacidad de beneficiar, tasa costo/beneficio, la edad, estatus social o “mérito” social.

También, se identifica que la priorización realizada por los clínicos, es algo arbitraria e incorpora sus propios valores y preferencias. Decidir la utilización de sistemas de priorización es algo complicado, ya que aún no se sabe si puede resultar mejor o peor. A pesar de que las decisiones realizadas por los médicos son subjetivas en muchos casos, algunas priorizaciones no se realizan de acuerdo a las necesidades clínicas, éstas son influenciadas, a veces por políticas gubernamentales por ejemplo.

Adicionalmente, a los elementos comunes en la priorización se agregan 3 elementos para considerar dentro de un sistema de priorización:

- **Tiempo de espera:** Aunque en varios casos se excluye el tiempo de espera en la priorización, esto es relativo de acuerdo a los objetivos del sistema. Por ejemplo, en enfermedades degenerativas es importante considerar el tiempo de espera.
- **Factores sociales:** Es importante de considerar en la prioridad de importancia, incluso se debe tratar sobre las decisiones clínicas (anularlas). Por ejemplo, la situación laboral, número de personas dependientes del paciente, etc.
- **Uso de los recursos:** Si se maximizan beneficios, es probable considerar el uso de los recursos. Se pueden calcular umbrales de puntuación en parte por los recursos.

4.3. Trabajos Internacionales

En (Santibáñez et al., 2007), realizan una programación por bloques, en donde se le asignan bloques a las distintas especialidades.

Mediante un problema de multiobjetivo de programación entera mixta realizan la programación por bloque de un cuerpo de ocho hospitales pertenecientes a la FHA (*Fraser Health Authority*), Canadá. El modelo matemático, presenta restricciones de disponibilidad de salas, disponibilidad de camas, preferencias de los médicos y lista de espera de pacientes.

El modelo es implementado en el software *General Algebraic Modeling System* (GAMS), y resuelto con las librerías de ILOG, CPLEX 9.0. Con tiempos de procesamientos para las instancias que van desde 12 a 48 hrs. Los resultados de la programación realizada en este estudio, muestra un incremento en la cantidad de pacientes a intervenir quirúrgicamente sin necesidad de aumentar los recursos del hospital.

En el estudio de (Fei, Meskens, & Chu, 2010), diseñan, una programación semanal de cirugías donde existe una estrategia de programación abierta. En este estudio se considera la utilización de sobretiempos y se realiza hincapié en la diferencia del costo operacional de las distintas salas quirúrgicas en jornada normal y en sobretiempo.

En este estudio se maximiza la utilización de salas quirúrgicas y minimiza el costo por sobretiempo. El problema se divide en dos fases: la primera, es el problema de planificar las cirugías semanalmente, este se resuelve asignando una fecha de cirugía para cada caso de cirugía. En la segunda, la programación de cada día es resuelta como un problema de programación diario.

La programación semanal es modelada como un problema de programación lineal entero mixto y se resuelve por el método de CGBH (*Column Generation Based Heuristic*). La programación diaria es transformada a un problema de programación de *flow shop* híbrido y resuelto con HGA (*Hybrid Genetic Algorithm*).

La programación, es realizada en un hospital de Bélgica, con tres instancias, las cuales se resuelven dentro de 345 a 1056 segundos. La instancia de mayor tamaño consiste en 131 pacientes en lista de espera y 31 médicos cirujanos disponibles. En los tres escenarios, se mejora la utilización de las salas quirúrgicas, los tiempos ociosos y se minimiza el uso de sobretiempo.

La publicación realizada por (Augusto, Xie, & Perdomo, 2010), investiga el impacto de permitir a un paciente recuperarse en una sala de operación cuando no existen camas en la sala de recuperación.

Se centran en una política de programación abierta, teniendo en consideración que el costo de utilización de 1h de la sala de operación es de 2 a 5 veces mayor que la utilización durante el mismo tiempo en una sala de recuperación. Además se consideran los transportes del

paciente durante el proceso completo. Sin embargo, se centran en un criterio de *makespan*, minimizando el tiempo de cierre de la sala de operaciones. El objetivo es atender a todos los pacientes posibles en un tiempo corto.

En el artículo se modela el problema como un *flow shop* híbrido de 4 etapas, donde cada recurso (transportadores, salas quirúrgicas y salas de recuperación) son máquinas paralelas idénticas. Las etapas son: (i) transporte de la recepción a la sala quirúrgica, (ii) cirugía y recuperación, (iii) las tareas de limpieza y (iv) transporte de la sala quirúrgica a la recepción.

El modelo matemático, consiste en minimizar el tiempo de completar los procesos para cada paciente, los recursos involucrados son: número de equipos de porteros idénticos en un periodo de tiempo, número de salas disponibles por periodo de tiempo, y número de camas de recuperación en un tiempo determinado.

Para resolver el problema, se utilizó una heurística basada en relajación Lagrangiana, donde se resuelven los subproblemas. Luego, para crear una solución factible (programación de cirugías) se utilizan 4 heurísticas distintas.

En este estudio, se compara el beneficio de hacer la recuperación de un paciente en una sala quirúrgica, mostrando que el beneficio es mayor cuando existe una gran sobrecarga sobre las camas de recuperación.

En el estudio de (Roland, Di Martinelly, Riane, & Pochet, 2010), se realiza una programación semanal de cirugías, teniendo en cuenta el recurso humano como gran parte del procedimiento. Para la programación de cirugías se identifican dos fases secuenciales. La primera etapa, de planificación, se identifica qué pacientes serán atendidos y posterior a esto una etapa de programación donde se establece el periodo de inicio de cada cirugía.

Para el problema de programación de cirugías electivas se utilizó un enfoque de proyectos (*Resource Constrained Project Scheduling, RCPSP*). Se modela el problema como un modelo de programación lineal entero mixto, y además, se resuelve con una metaheurística de algoritmos genéticos. Se minimizan los costos de utilización de las salas de operación y el pago de sobretiempos.

Se realiza el estudio en un hospital Belga, utilizando instancias con 80 pacientes en lista de espera, 33 cirujanos, 7 salas quirúrgicas, 5 días de programación y cada día es dividido en 48 periodos de tiempo. Además, comparan los resultados obtenidos en un día de programación entre el modelo de programación lineal entero y la heurística propuesta. Los resultados obtenidos en el modelo matemático fueron encontrados en un tiempo de 1800 segundos en cada instancia, encontrando el óptimo en cada una de ellas. En cambio, la heurística basada en algoritmos genéticos encontró los resultados en tiempos computacionales de 135-322 segundos, obteniendo el óptimo o valores muy cercanos a él.

En (Riise & Burke, 2011), desarrollan un algoritmo metaheurístico multiobjetivo (tiempo de espera, sobretiempo de cirujanos, tiempo espera niños en las mañanas). El algoritmo utiliza vecindarios de reubicación y dos-intercambios para una búsqueda local (VND). Los resultados son comparados con las programaciones realizadas en un hospital de Noruega.

En (Liu, Chu, & Wang, 2011), se exhibe la programación de salas quirúrgicas con una estrategia de programación abierta en donde los cirujanos pueden usar todos los bloques de horarios disponibles. Se genera una programación de cirugías para maximizar el uso eficiente de las salas quirúrgicas y minimizar el costo de tiempo extra en las salas quirúrgicas, mostrando en los resultados computacionales la eficiencia del algoritmo para instancias grandes, el cual muestra siempre una solución factible. Una de las ventajas del algoritmo utilizado es que puede ser usado en metaheurísticas o *Branch & Bound* para generar una de las soluciones iniciales.

Los tiempos computacionales del algoritmo propuesto en una instancia de 160 cirugías para una estrategia abierta van desde los 27 a 108 segundos, mejorando notablemente los resultados en tiempos computacionales en comparación con otro algoritmo propuesto anteriormente por los mismos autores.

En (Agnetis et al., 2012), se estudian las políticas de planificación de salas quirúrgicas para pacientes electivos. Se hace énfasis en las políticas de planificación a largo plazo para determinar el Plan Maestro de Cirugías (MSS, por sus siglas en inglés *Master Surgical Schedule*) a lo largo del año.

En el estudio, se otorgan herramientas para evaluar los beneficios y sus derivados de un MSS dinámico, investigando el efecto de varias políticas de MSS en la calidad de los planes quirúrgicos.

Los principales tipos de restricciones propuestos en el modelo MIP son:

- Disciplinas para salas quirúrgicas: Ciertas disciplinas pueden ser realizadas en salas quirúrgicas con condiciones específicas, debido al tamaño y/o equipamiento.
- Límites en el paralelismo disciplinario: No más de k intervenciones de una cierta especialidad pueden ser realizadas al mismo tiempo por restricciones de personal y de equipamiento.
- Restricciones de intervenciones por especialidad: Balance de la carga laboral.
- Reservaciones de salas quirúrgicas: El hospital decide si hay cirugías reservadas para cierto día y que no se pueden atrasar.

La priorización de cirugías, se realiza por una puntuación (*score*) y también se introducen en el estudio 3 modelos de decisión de una semana (5 días): *Fixed model*, *Bounded model*, *Flexible model*.

El estudio se realiza en un hospital de tamaño mediano en Empoli, Italia y se evalúa el alcance que podría tener en hospitales de mayor tamaño. En el estudio de caso realizado, el MSS propuesto mejora en términos de eficiencia la utilización de recursos y el desempeño en términos de fechas en que las cirugías deben estar realizadas.

En (Vijayakumar et al., 2013), realizan un *case scheduling* en un hospital financiado con fondos públicos en Dayton, Estados Unidos, el cual consiste en asignar a cada caso quirúrgico un periodo de tiempo en el día, y día de la semana dependiendo de la disponibilidad de los recursos como salas quirúrgicas, cirujanos, enfermeras y equipos médicos.

Se aborda el problema como un *dual-bin packing*. Un *multi-bin* (varios días), distinto tamaño (distintos horarios en los turnos), multidimensional (múltiples recursos). Es así como en este trabajo se modela el problema de *case scheduling* como un problema de programación lineal entero mixto, resuelto con una heurística basada en *First Fit Decreasing* (FFD). El modelo

tiene como objetivo maximizar la cantidad de pacientes atendidos a lo largo de los días y además respetar la prioridad ponderando cada paciente con los atributos de cada cirugía.

El hospital utilizado como caso de estudio consta de 7 salas quirúrgicas, 10 especialidades, 32 médicos cirujanos y realizan las programaciones de cirugías electivas manualmente. El estudio muestra una reducción del 20% en los días utilizados para intervenir quirúrgicamente a la misma cantidad de pacientes y una mejora del 20% en la utilización de salas quirúrgicas, con tiempos computacionales en la instancia de mayor tamaño (100 pacientes, 5 salas quirúrgicas, 5 cirujanos) con la heurística propuesta basada en FFD, de 120 segundos. Estos resultados muestran una notable mejora en tiempos computacionales en comparación con el modelo de programación lineal propuesto en este estudio, ya que para una instancia de 15 pacientes, 32 periodos de tiempo por día, 5 salas quirúrgicas, 5 cirujanos, se resolvió el modelo en 12 horas para una programación de cirugías electivas de 2 días.

En (Meskens, Duvivier, & Hanset, 2013), se realiza una revisión bibliográfica de trabajos realizados relacionados al tema en donde analizan los tipos de restricciones de cada modelo, proponiendo un modelo genérico, el cual incluye la mayoría de los tipos de restricciones según los autores.

Dado el gran número de restricciones que presenta el problema de programación de cirugías electivas, identifican la programación por restricciones como una herramienta adecuada para abordar el problema. En el grupo de restricciones se destaca un grupo de restricciones de afinidad en el *staff* médico, apuntando a un mejor desempeño obtenido con una gran afinidad en el *staff* médico (solo tratado en una publicación anterior según los autores).

Proponen un modelo multiobjetivo con características modulares, en donde se agrupan restricciones creando 5 diferentes módulos: programación en bloques (disponibilidad), cirugías, recursos humanos, materiales y afinidad entre el *staff*. Se consideran recursos estériles que pueden usarse solo 2 veces al día. Los objetivos del modelo son minimizar el *makespan*, minimizar el sobretiempo y maximizar la afinidad del *staff*.

Mediante instancias reales obtenidas de un hospital universitario de Bélgica, prueban distintas configuraciones del modelo propuesto y miden el desempeño del modelo en cada

configuración, mostrando mejores tiempos computacionales en configuraciones que poseen más restricciones donde por ejemplo en una instancia donde en un día esperan 16 pacientes, existen 6 cirujanos, 4 salas quirúrgicas, 4 anestelistas, 6 enfermeras y 4 camas de recuperación, el tiempo computacional fue de 1462 segundos.

La publicación de (Fügener, Hans, Kolisch, Kortbeek, & Vanberkel, 2014), Se concentra problemas del MSS a nivel táctico y cómo afecta a los otros servicios relacionados (ICU, PACU). Se apunta a minimizar el costo de unidades relacionadas, nivelando la demanda de camas y reduciendo las solicitudes de camas de fin de semana. Realizan un modelo estocástico con el fin de realizar un MSS de acuerdo a los posibles caminos que toman los pacientes a los diferentes servicios con datos obtenidos de un hospital alemán. Por la complejidad del problema, además de encontrar el óptimo con *Branch and Bound*, se proponen dos estrategias heurísticas para resolver el problema: función objetivo exacta y método de solución heurístico (IIH, 2-Opt y *Simulated Annealing*) y función objetivo aproximada y método de solución exacto (solo valores esperados y una combinación de valores esperados y varianzas).

4.4. Análisis de la revisión bibliográfica del problema

Este tipo de problemas, es considerado como un *NP-Hard*, porque existe una alta combinatoria. Por esta razón en algunos artículos se modela bajo ciertos supuestos, dejando fuera algunos tipos de restricciones.

En la Tabla 4.2 se consideran las restricciones claves según este trabajo para el problema de programación de cirugías electivas presentadas en las publicaciones revisadas. En la Tabla 4.1 se muestra la leyenda de la Tabla 4.2 junto a los tipos de restricciones que son consideradas.

Mientras la mayoría de los artículos presentan el problema con una formulación combinatoria de optimización, los autores plantean distintos enfoques en que identifican el impacto directo del cambio de configuración de un elemento o más sobre el problema, como por ejemplo: el uso de las salas quirúrgicas como camas de recuperación, la afinidad del *staff* médico para evaluar el desempeño en las cirugías, etc. (Augusto et al., 2010; Meskens et al., 2013).

En gran parte de la literatura revisada plantean el problema de programación de cirugías electivas con un modelo de programación lineal entero mixto (MIP), pero realizan una distinción a la hora de definir el modelo, es decir, existen enfoques de proyectos (Roland et al., 2010), programación por restricciones (Fei et al., 2010), *flow shop* (Latorre, 2011), etc.

TABLA 4.1. LEYENDA TABLA 4.2.

| Consideración | • |
|------------------------------------|----------|
| Disponibilidad Cirujanos | A |
| Disponibilidad Enfermeras | B |
| Disponibilidad Anestesistas | C |
| Disponibilidad Salas Quirúrgicas | D |
| Compatibilidad Salas Quirúrgicas | E |
| Compatibilidad Cirujano | F |
| Sobretiempo | G |
| Prioridad Paciente | H |
| Insumos Médicos | I |
| Camas de Hospitalización | J |
| Afinidad del <i>Staff</i> Médico | K |
| Carga Laboral | L |
| Disponibilidad Parcial de Recursos | M |

TABLA 4.2. TIPOS DE RESTRICCIONES DEL PROBLEMA CONSIDERADAS EN LA LITERATURA

| Publicación | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| (Fei et al., 2010) | • | | | • | | • | | | • | • | | | • |
| (Augusto et al., 2010) | • | | | • | • | | | | | | | | |
| (Roland et al., 2010) | • | | | • | • | • | | • | | • | | | |
| (Latorre,2011) | • | | | • | • | • | | | | | | | |
| (Liu et al., 2011) | • | • | | • | | | | | | • | | | • |
| (Riise & Burke, 2011) | • | | | • | • | • | • | • | | | | | |
| (Pradenas et al., 2012) | • | • | • | • | • | • | | • | | • | | | |
| (Wolff, Durán & Rey, 2012) | • | | | • | • | • | • | • | | | | • | • |
| (Pradenas Rojas & Matamala Vergara, 2012) | • | • | • | • | • | • | | • | • | • | • | | |
| (Meskens et al., 2013) | • | | | • | • | • | • | | | | | • | |
| (Vijayakumar et al., 2013) | • | • | • | | • | | | | • | | | | • |

En cuanto a los modelos propuestos, en la literatura se encuentran diferentes realidades donde se aplica cada estudio. Sin embargo, es necesario rescatar los conjuntos de restricciones que son considerados más importantes para este trabajo, ya sea en lo que refleja la realidad local, como en lo que es mencionado reiterativamente en la literatura.

Ya que el problema de programación y planificación de cirugías electivas es NP-Hard (Vijayakumar et al., 2013), existe un gran número de publicaciones en donde se opta por métodos de resolución distintos a los métodos exactos, ya que son a lo menos NP-Complejos. Algunos autores optan por heurísticas (Augusto et al., 2010; Fei et al., 2010; Liu et al., 2011; Pradenas et al., 2012; Vijayakumar et al., 2013) o metaheurísticas (Fei et al., 2010; Pradenas Rojas & Matamala Vergara, 2012; Roland et al., 2010) para una resolución más eficiente del problema.

En la Tabla 4.3 se resumen los métodos de resolución usados por las publicaciones revisadas, en donde se pueden encontrar heurísticas, metaheurísticas y métodos exactos.

Dentro de la literatura se destacan conjuntos de restricciones utilizadas comúnmente tales como la disponibilidad de cirujanos, disponibilidad del staff médico y la compatibilidad de cirujanos con cada caso quirúrgico. Sin embargo, existen aspectos no tan profundizados, pero considerados claves ya que se presentan como una limitante en la programación de cirugías electivas. En estos aspectos se encuentra la disponibilidad de salas quirúrgicas, el uso de sobretiempo, la disponibilidad parcial de recursos (staff médico) y recursos no renovables inmediatamente (insumos estériles). Los conjuntos de restricciones encontrados en la literatura son claramente relacionados con los supuestos realizados en cada estudio en particular. Los supuestos más frecuentes encontrados en la revisión bibliográfica, consisten principalmente en la disponibilidad permanente del staff médico, el uso de salas quirúrgicas como máquinas multifuncionales, la asignación de un cirujano para cada caso quirúrgico (sin considerar médicos asistentes), entre otros.

De las publicaciones revisadas, se puede detectar un patrón en las investigaciones con estudios de caso, donde el problema se modela y se implementa en instancias pequeñas debido a los altos tiempos de computación, estos tiempos elevados se presentan como una limitante al

momento de llevarlo a una aplicación, ya que se necesita que la programación de cirugías electivas se realice en tiempos razonables. Por esta razón, gran parte de los autores de la revisión bibliográfica resuelven el problema con heurísticas y metaheurísticas, que entregan resultados en tiempos considerablemente menores y de buena calidad a pesar de que no siempre encuentran el óptimo. Esta tendencia se da en cierto grado por los grandes avances computacionales, que permiten resolver problemas altamente combinatorios en tiempos del orden de segundos.

Las metaheurísticas muestran los resultados más prometedores, y dentro de la revisión de la literatura se destacan los algoritmos genéticos como los más estudiados y con resultados de alta calidad. Sin embargo, existen metaheurísticas que aún no han sido tan profundizadas para este problema, tales como *Simulated Annealing*, *Tabu Search*, y metaheurísticas de búsquedas variables, entre otros.

TABLA 4.3. MÉTODO DE RESOLUCIÓN UTILIZADOS EN LA LITERATURA.

| Publicación | Método de Resolución |
|---|--|
| Santibáñez et al., 2007 | Modelamiento Matemático(CPLEX/GAMS) |
| Fei et al., 2010 | Heurística basada en Generación de Columnas/Algoritmo Genético Híbrido |
| Augusto et al., 2010 | Heurística basada en Relajación Lagrangiana |
| Roland et al., 2010 | Metaheurística basada Algoritmos Genéticos |
| Latorre, 2011 | Heurística Constructiva |
| Riise and Burke, 2011 | Metaheurística basada en ILS y VND |
| Liu et al., 2011 | Heurística basada en Programación Dinámica |
| Pradenas et al., 2012 | Multi-knapsack/Chronological Backtracking Heuristic |
| Pradenas Rojas & Matamala Vergara, 2012 | Algoritmos Genéticos |
| Wolff, Durán & Rey, 2012 | Modelamiento Matemático(CPLEX)/Backtracking |
| Vijayakumar et al., 2013 | Heurística basada en First Fit Decreasing |
| Meskens et al., 2013 | CHOCO 2.1.0 Satisfacción de Restricciones |
| Fügener et al. 2014 | Heurísticas IIH, 2-OPT, SA |

En (Cartes, 2014), el modelo de programación lineal entero propuesto se implementaron restricciones de: disponibilidad de médicos, disponibilidad de salas quirúrgicas, uso de sobretiempo en la programación de las salas, priorización de pacientes, disponibilidad de camas

de hospitalización, entre otros. Estas restricciones y objetivos serán profundizados en el Capítulo 5.

Desde la perspectiva del uso de metaheurísticas, el algoritmo GRASP propuesto y explicado con más detalle en el Capítulo 6 de este informe, otorga un tratamiento del problema de forma moderna, utilizando una metaheurística relativamente nueva y sin utilizar para el problema tratado, la cual no se encuentra en ninguno de los artículos revisados en el estado del arte, en comparación con las metaheurísticas comúnmente utilizadas y más clásicas como: Algoritmos Evolutivos, *Tabu Search*, *Simulated Annealing*.



Capítulo 5. Modelamiento del problema de cirugías electivas

5.1. Introducción

En este capítulo se presenta el modelo de programación lineal entero para el problema de programación de cirugías electivas propuesto en un trabajo anterior (Cartes, 2014). El modelo de programación lineal entero es presentado en este trabajo para realizar una comparación con el algoritmo GRASP en términos de desempeño.

Este capítulo presenta las consideraciones particulares del problema y una descripción del modelo: índices, parámetros, variables de decisión, funciones objetivos y restricciones.

5.2. Consideraciones particulares

Aunque este modelo trata de representar de manera completa la programación de cirugías electivas, se realizaron ciertos supuestos:

- Los pacientes están siempre disponibles para ser intervenidos, a diferencia de la publicación de (Wolff, Durán, & Rey, 2012).
- La prioridad de los pacientes se mantiene constante durante el horizonte de planificación.
- A cada cirugía se le asigna un solo médico. En el caso de necesitar un médico secundario este puede ser asignado posteriormente.
- La asignación del equipo (*staff*) médico para cada cirugía es asignado posteriormente.

Una vez comenzada la cirugía, no puede ser interrumpida. Es decir, una vez que comienza una cirugía dada, debe terminarse. Para el tiempo de duración estimado de cada cirugía es necesario considerar los tiempos de preparación y limpieza de la sala a utilizar, además del tiempo de cirugía estimado.

En el contexto de la priorización de pacientes electivos, la priorización recae sobre el tomador de decisiones, otorgándole la ponderación o método de su preferencia a la prioridad del paciente.

5.3. Modelo de programación lineal entero

5.3.1 Índices

Los índices utilizados por el modelo de programación lineal entero son expuestos en la Tabla 5.1

TABLA 5.1. ÍNDICES DEL MODELO MATEMÁTICO

| Índices | |
|-------------------------|-------------------|
| $p \in \{1, \dots, P\}$ | Paciente |
| $m \in \{1, \dots, M\}$ | Médico |
| $s \in \{1, \dots, S\}$ | Sala Quirúrgica |
| $t \in \{1, \dots, T\}$ | Periodo de tiempo |
| $i \in \{1, \dots, D\}$ | Día |

5.3.2 Parámetros

Los parámetros propuestos para el modelo de programación lineal entero de programación de cirugías electivas son mostrados en la Tabla 5.2.

TABLA 5.2. PARÁMETROS UTILIZADOS POR EL MODELO MATEMÁTICO

| Parámetros | |
|---------------|---|
| Pri_p | Prioridad del paciente p . |
| CST_s | Factor asociado al costo de utilizar la sala quirúrgica s en sobretiempo. |
| Cam_i | Camas disponibles en el día i . |
| TO_p | tiempo de operación de la cirugía del paciente p . |
| $med_{m,t,i}$ | 1 si el médico m trabaja durante el periodo t el día i , 0 en otro caso. |
| $sal_{s,t,i}$ | 1 si la sala quirúrgica s está disponible durante el periodo t el día i , 0 en otro caso. |
| FM_p^m | 1 si el médico m puede operar al paciente p , 0 en otro caso. |
| FS_p^s | 1 si la sala s cumple las condiciones para atender al paciente p , 0 en otro caso. |
| nE_i^t | Número de enfermeras compatibles disponibles en el periodo t en el día i . |
| nA_i^t | Número de anestelistas disponibles en el periodo t en el día i . |
| M_m | Cantidad máxima de cirugías para el médico m en el horizonte de programación. |
| π | Conjunto de periodos de tiempo que pertenecen al sobretiempo |

5.3.3 Variables de decisión

Las variables de decisión son mostradas en la Tabla 5.3.

TABLA 5.3. VARIABLES DE DECISIÓN UTILIZADAS EN EL MODELO MATEMÁTICO

| Variables de Decisión | Descripción |
|------------------------------|---|
| $a_{p,s,m,t,i}$ | 1 si el paciente p se atiende en la sala s con el médico m en el periodo t el día i , 0 en otro caso. |
| x_t^s | 1 si la sala quirúrgica s es utilizada en sobretiempo durante el periodo t , 0 en otro caso. |
| $o_{p,s,i}$ | 1 si el paciente p es asignado a la sala quirúrgica s el día i , 0 en otro caso. |
| $y_{p,s}$ | 1 si el paciente p se opera en la sala quirúrgica s , 0 en otro caso. |
| $l_{p,m}$ | 1 si el paciente p se opera con el médico m , 0 en otro caso. |
| t_p^e | Periodo de finalización de la cirugía del paciente p . |
| t_p^s | Periodo donde empieza la cirugía del paciente p . |

5.3.4 Funciones objetivo

$$\text{Max } \sum_{p \in P, s \in S, i \in I} \text{Pri}_p \cdot o_{p,s,i} \quad (5.1)$$

$$\text{Min } \sum_{s \in S, t \in T} \text{CST}_s \cdot x_t^s \quad (5.2)$$

Las relaciones, (5.1) y (5.2) son los objetivos del modelo, donde (5.1) maximiza la prioridad de los pacientes intervenidos quirúrgicamente durante el periodo de programación y (5.2) minimiza la cantidad de sobretiempo utilizado para cada sala quirúrgica durante el mismo periodo.

5.3.5 Restricciones del modelo

Sujeto a:

$$\sum_{s \in S, m \in M, i \in I} a_{p,s,m,t,i} \leq 1 \quad \forall p, t \quad (5.3)$$

$$\sum_{p \in P, m \in M} a_{p,s,m,t,i} \leq 1 \quad \forall s, t, i \quad (5.4)$$

$$\sum_{p \in P, s \in S} a_{p,s,m,t,i} \leq 1 \quad \forall m, t, i \quad (5.5)$$

$$\sum_{s \in S, i \in I} o_{p,s,i} \leq 1 \quad \forall p \quad (5.6)$$

$$\sum_{m \in M, t \in T} a_{p,s,m,t,i} \leq T * o_{p,s,i} \quad \forall p, s, i \quad (5.7)$$

$$\sum_{s \in S, i \in I} o_{p,s,i} = \sum_{m \in M} l_{p,m} \quad \forall p \quad (5.8)$$

$$\sum_{i \in I} o_{p,s,i} = \sum_{p \in P, s \in S} y_{p,s} \quad \forall p, s \quad (5.9)$$

$$\sum_{s \in S, t \in T, i \in I} a_{p,s,m,t,i} = T O_p * l_{p,m} \quad \forall p, m \quad (5.10)$$

$$\sum_{s \in S} a_{p,s,m,t,i} \leq med_{m,t,i} * l_{p,m} \quad \forall p, m, t, i \quad (5.11)$$

$$\sum_{m \in M} a_{p,s,m,t,i} \leq sal_{s,t,i} * y_{p,s} \quad \forall p, s, t, i \quad (5.12)$$

$$t_p^s \leq t * \sum_{s \in S, m \in M, i \in I} a_{p,s,m,t,i} + T(\sum_{s \in S, m \in M, i \in I} a_{p,s,m,t,i}) \quad \forall p, t \quad (5.13)$$

$$t_p^e \geq (t + 1) \sum_{s \in S, m \in M, i \in I} a_{p,s,m,t,i} \quad \forall p, t \quad (5.14)$$

$$\sum_{m \in M, s \in S, t \in T, i \in I} a_{p,s,m,t,i} = t_p^e - t_p^s \quad \forall p \quad (5.15)$$

$$\sum_{s \in S, t \in T, i \in I} a_{p,s,m,t,i} \leq FM_p^m (t_p^e - t_p^s) \quad \forall p, m \quad (5.16)$$

$$\sum_{m \in M, t \in T, i \in I} a_{p,s,m,t,i} \leq FS_p^s (t_p^e - t_p^s) \quad \forall p, s \quad (5.17)$$

$$\sum_{p \in P} l_{p,m} \leq M_m \quad \forall m \quad (5.18)$$

$$\sum_{p \in P, s \in S} o_{p,s,i} \leq Cam_i \quad \forall i \quad (5.19)$$

$$\sum_{p \in P, s \in S, m \in M} a_{p,s,m,t,i} \leq nE_i^t \quad \forall t, i \quad (5.20)$$

$$\sum_{p \in P, s \in S, m \in M} a_{p,s,m,t,i} \leq nA_i^t \quad \forall t, i \quad (5.21)$$

$$\sum_{p \in P, m \in M} a_{p,s,m,t,i} = x_t^s \quad \forall s, i ; t \in \pi = \{t_{st}\} \quad (5.22)$$

$$a_{p,m,s,t,i} \in \{0,1\} \quad \forall p, m, s, t, i \quad (5.23)$$

$$o_{p,s,i} \in \{0,1\} \quad \forall p, i \quad (5.24)$$

$$t_p^f, t_p^e \in \{1,2, \dots, T\} \quad \forall p \quad (5.25)$$

$$l_{p,m} \in \{0,1\} \quad \forall p, m \quad (5.26)$$

$$y_{p,s} \in \{0,1\} \forall p, s \quad (5.27)$$

$$x_t^s \in \{0,1\} \forall s, t \quad (5.28)$$

Las restricciones (5.3), indican que en cualquier periodo, un paciente puede ser asignado a lo más a un cirujano y sala quirúrgica en un día dado. Las restricciones (5.4) establecen, que a lo más un cirujano puede ser programado en una sala quirúrgica en un día y periodo dado. El conjunto de restricciones (5.5) establece, que un cirujano puede realizar como máximo una cirugía en un día y periodo dado. Las restricciones (5.6) establecen, que un paciente puede ser programado solo una vez a lo largo del periodo de programación. Las restricciones (5.7), (5.8) y (5.9) permiten, derivar o interconectar las variables de decisión $o_{p,s,i}$, $l_{p,m}$ y $y_{p,s}$; permitiendo además el uso de éstas en las funciones objetivos. Las restricciones (5.10) determinan, que la suma de los bloques de tiempo utilizados en la operación de un paciente dado, debe ser igual al tiempo estimado para esa operación. Los grupos de restricciones (5.11) y (5.12) son restricciones de disponibilidad, donde no se puede programar una cirugía si es que no existe médico ni sala disponible en un periodo y día dado. Los conjuntos de restricciones (5.13) y (5.14) determinan, los periodos de comienzo y término de cirugías respectivamente. Las restricciones (5.15) establecen, que el número de periodos de tiempo requeridos por una cirugía dada debe ser igual a la diferencia entre el bloque de tiempo de final de cirugía y el de inicio de cirugía. Las restricciones (5.16) y (5.17) son restricciones de compatibilidad o factibilidad, donde no se puede programar una cirugía si el cirujano no cumple los atributos para atender al paciente y la sala no cumple con los requerimientos necesarios para llevar a cabo una cirugía dada, respectivamente. Las restricciones (5.18) aseguran, que un médico dado no pueda realizar más allá de un número máximo de cirugías establecidas para él, estas restricciones permiten controlar la carga laboral de los cirujanos. Las restricciones (5.19) aseguran, que una cirugía puede ser programada siempre que existan camas de hospitalizados disponibles. Las restricciones (5.20) y (5.21) son restricciones de disponibilidad de enfermeras y anestesistas respectivamente, en donde una cirugía no puede ser programada si no existe una enfermera y anestesista disponible. Las restricciones (5.22) tratan del uso de sobretiempo y establecen que si una cirugía es programada en los periodos de sobretiempo (π), la variable x_t^s tomará el valor

1 para un día y periodo dado. Las restricciones (5.23)-(5.28) son restricciones de binariedad e integralidad de las variables.

5.4. Implementación

El modelo matemático de optimización anteriormente presentado, es implementado en el lenguaje de programación *Python*, en el ambiente de programación *Spyder*, para resolverlo así con GNU *Linear Programming Kit* (GLPK), versión 4.45. Se utilizó el *presolver* de GLPK, cual es un *solver* de programación entera que utiliza un algoritmo de *Branch and Cut* en un procesador Intel Core i5 de 2.53 GHz, 4GB de memoria RAM, en la distribución de Linux: Ubuntu 12.04 de 64 bits.

El GNU *Linear Programming Kit* es un software que forma parte del proyecto GNU y tiene el propósito de resolver problemas de programación lineal de gran escala (LP), programación lineal entera mixta (MIP) y otros problemas relacionados. Es un conjunto de rutinas escritas en ANSI C y organizadas como una librería.

Al momento de la implementación se utilizó la *callable library* de GLPK para implementar el modelo matemático, para esto se necesitaron los *bindings* Python-GLPK.

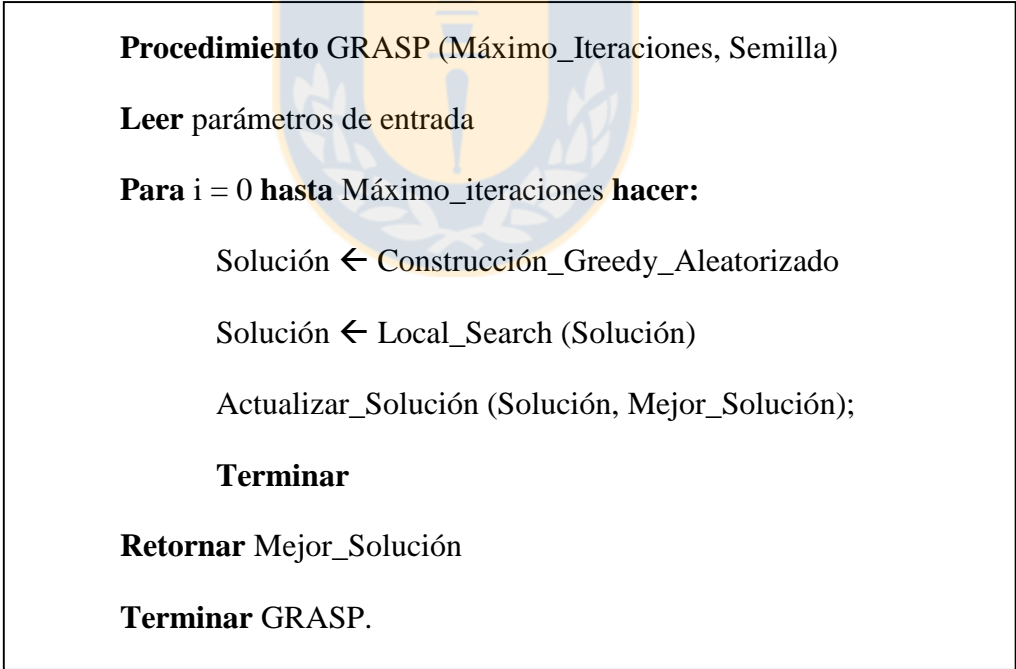
Además, se utilizó el *solver* CPLEX *interactive optimizer* 12.5.0.0 en OS X El Capitán versión 10.11.4, en un procesador Intel Core i7 de 2.9 GHz, con 8 GB de memoria RAM, el cual resolvió los archivos LP exportados desde los modelos realizados en GLPK. Para realizar una razón entre los dos computadores, se ejecutó un mismo programa sin dependencias de *solvers* ni librerías. La razón del computador en el cual se utilizó CPLEX es de 0.74.

Dado que el modelo matemático es multiobjetivo, se utilizó el método de la suma lineal de pesos para el tratamiento del modelo. Se utilizó este método, ya que el problema presenta mayor importancia relativa en el objetivo 1 (ecuación 5.1). Este método permite al tomador de decisiones otorgar los ponderadores de su preferencia para el modelo. Así, los valores de los ponderadores fueron fijados en $\beta = 0.7$ para el objetivo 1 (ecuación 5.1) y $\gamma = 0.3$ para el objetivo 2 (ecuación 5.2) finalmente.

Capítulo 6. Algoritmo GRASP para el problema de programación de cirugías electivas y experimentos

6.1. Introducción

La metaheurística GRASP (*Greedy Randomized Adaptive Search Procedures*) (Resende & Ribeiro, 2003), consta de dos etapas. Primero, existe una etapa de construcción (Sección 6.4) en donde el algoritmo GRASP crea una solución factible en base a un algoritmo Greedy aleatorio. Posteriormente, los vecindarios de la solución factible generada en la etapa de construcción son recorridos hasta encontrar el óptimo local para dicha solución, esta etapa se conoce como búsqueda local (Sección 6.5), así estas etapas se repiten hasta que se alcanza el número máximo de iteraciones, *Max_iter*. El algoritmo GRASP se describe de forma global en la Figura 5.



```
Procedimiento GRASP (Máximo_Iteraciones, Semilla)  
Leer parámetros de entrada  
Para  $i = 0$  hasta Máximo_iteraciones hacer:  
    Solución  $\leftarrow$  Construcción_Greedy_Aleatorizado  
    Solución  $\leftarrow$  Local_Search (Solución)  
    Actualizar_Solución (Solución, Mejor_Solución);  
Terminar  
Retornar Mejor_Solución  
Terminar GRASP.
```

FIGURA 5. PSEUDOCÓDIGO ALGORITMO GRASP

6.2. Algoritmo GRASP

En primer lugar, el algoritmo propuesto genera todos los elementos posibles con todas las posibles combinaciones de acuerdo a los parámetros, dadas las características de cada instancia. Es decir, para cada paciente se crean los elementos con todos los médicos y salas compatibles y en todos los periodos de programación de acuerdo a la duración de su cirugía. Obtenidos todos los elementos, se crea una lista ordenada: de forma ascendente de acuerdo a la ID del paciente, de forma descendente de acuerdo a la prioridad del paciente y de forma ascendente de acuerdo al periodo de programación (es decir, las cirugías programadas al principio del día irán en primer lugar hasta llegar a las cirugías programadas en los últimos periodos).

6.2.1 Representación de la solución

La representación de la solución del algoritmo GRASP propuesto, se representa con números enteros, fraccionales y binarios. La información entregada en cada elemento (de izquierda a derecha, como se muestra en la Figura 6) de la solución es: Prioridad del paciente, Duración cirugía del paciente, ID del paciente, Costo de sobretiempo asociado a la sala de la programación, Día de programación, Compatibilidad médico-paciente, Compatibilidad sala-paciente y Programación de la cirugía en determinados periodos de tiempo.

Como ejemplo se observa la Figura 7, que muestra un elemento descrito de la siguiente manera: El paciente 1, posee una prioridad de ser atendido de un 0.20, con una duración de 4 periodos de tiempo en su cirugía asociada, donde al ser operado en periodo de sobretiempo tiene un ponderador de costo de sobretiempo de 0.14 en la sala programada. El paciente 1, en este elemento puede ser intervenido quirúrgicamente solo con el médico 1, en la sala 3 y su programación en esta solución está destinada a los 4 primeros periodos de tiempo del día 3.

| | | | | | | | |
|---------------|--------------|-------------|-----------|-----|---------------------------------------|----------|---|
| Prioridad p | Duración p | ID Paciente | CST s/p | Día | Asig m | Asig s | Programación en periodos de acuerdo a la duración |
| | | | | | Largo total médicos Largo total salas | | Largo total periodos |

FIGURA 6. ESTRUCTURA DE LOS ELEMENTOS EN EL ALGORITMO GRASP

| | | | | | | | |
|------|---|---|------|---|-------------|-----------|---------------|
| 0.20 | 4 | 1 | 0.14 | 3 | 1,0,0,0,0,0 | 0,0,1,0,0 | 1,1,1,1,0,0,0 |
|------|---|---|------|---|-------------|-----------|---------------|

FIGURA 7. EJEMPLO DE UNA REPRESENTACIÓN DE UN ELEMENTO EN EL ALGORITMO GRASP

6.2.2 Función *Fitness*

La función *fitness*, de manera similar que en los objetivos del modelo de programación lineal entero, evalúa cada elemento con el fin de maximizar la prioridad de pacientes atendidos y minimizar el uso de sobretiempo en las salas quirúrgicas.

En el caso de la función *fitness*, cada elemento es evaluado con la ecuación 6.1. Es decir, calcula la diferencia ponderada por β y γ entre la prioridad del paciente de cada elemento y el costo de sobretiempo asociado (en el caso que corresponda). Para este caso, se utilizaron los mismos valores de β y γ presentados en el trabajo anterior (Cartes, 2014), con el objetivo de comparar los valores de la función *fitness* del algoritmo con la función objetivo del modelo de programación lineal entero. Cabe mencionar que estos ponderadores pueden ser modificados de acuerdo a las preferencias del tomador de decisiones

En términos de implementación del algoritmo la función *fitness*, es la suma del primer parámetro de los elementos (Prioridad del paciente) programados por el algoritmo GRASP. En términos prácticos, el parámetro de costo de sobretiempo asociado a la sala quirúrgica (CST_s) será sustraído al parámetro de prioridad del paciente (Pri_p) cuando utilice al menos un periodo de sobretiempo en su programación.

$$\beta \left(\sum_{s \in S, i \in I} Pri_p \cdot o_{p,s,i} \right) - \gamma \left(\sum_{s \in S, t \in T} CST_s \cdot x_t^s \right) \quad (6.1)$$

6.2.3 Etapa Constructiva

Con la lista de elementos ordenada, como se explica en la sección 6.1, se forma una solución factible mediante un algoritmo constructivo greedy pseudo-aleatorio. Un parámetro α define los elementos candidatos a ser considerados en la solución (RCL). Aleatoriamente, se selecciona un elemento de la RCL para ser programado, luego se elimina de la lista de candidatos todos los elementos que pueden generar infactibilidad en la solución parcial actual, ya sea porque programa al mismo paciente, por superar la cantidad de cirugías permitidas del médico o incompatibilidad de horarios con el médico y/o sala. Este procedimiento se repite hasta que no hay candidatos en la lista, es decir, la solución está completa. Para más detalle véase el pseudocódigo de la Figura 8.

6.2.4 Etapa de Búsqueda Local

Como segunda etapa, la búsqueda local (pseudocódigo Figura 9) se realiza mediante dos vecindarios establecidos: Relocate y Mejor candidato no programado. El vecindario Relocate, basado en el vecindario propuesto en (Riise & Burke, 2011), consiste en eliminar un elemento de la solución parcial al azar, y posteriormente buscar entre los elementos no programados (incluido el mismo elemento eliminado) un elemento factible de programar. El vecindario Mejor candidato no programado, consiste en eliminar de la solución el elemento que contenga al paciente con menor prioridad, luego los elementos no programados (incluido el mismo elemento eliminado) se ordenan de forma descendente por prioridad y se busca en dicho conjunto de elementos, algún elemento o elementos factible(s) y con mayor prioridad para programar dada la solución parcial.

Procedimiento Construcción_Greedy_Pseudoaleatorio (Semilla)

Solución $\leftarrow \phi$

Mientras Solución no esté completa **o** Lista de elementos no esté vacía
hacer:

Elegir un elemento s al azar de los $\alpha * \text{Largo_Lista_Elementos}$

Si s es factible **hacer:**

Solución \leftarrow Solución $\cup \{s\}$

Actualizar_Parámetros

Terminar

Retornar Solución

Terminar Construcción_Greedy_Pseudoaleatorio.

FIGURA 8. PSEUDOCÓDIGO ETAPA CONSTRUCTIVA DEL ALGORITMO GRASP

Procedimiento Búsqueda_Local (Solución)

Mientras Solución no esté completa **hacer:**

Solución \leftarrow Relocate (Solución)

Solución \leftarrow Mejor_candidato_no_programado

Retornar Solución

Terminar

FIGURA 9. PSEUDOCÓDIGO ETAPA DE BÚSQUEDA LOCAL ALGORITMO GRASP

6.3. Análisis del peor de los casos

El algoritmo complejidad del algoritmo viene dada por la lista de elementos, la cual determina las instrucciones que deberán realizarse en el algoritmo.

El algoritmo GRASP posee una complejidad $O(Max_iter \times pacientes \times médicos \times salas \times periodos \times días)$, el cual genera una lista que se recorre las n iteraciones del algoritmo, con todos los pacientes y todas las posibles combinaciones de: asignación de médicos, asignación de salas, asignación en la programación de periodos y asignación en la programación de días. Donde el peor de los casos sería una lista de pacientes que cumpla con todas las factibilidades, es decir, sea compatible con todos los médicos, salas, y pueda ser programado en cualquier periodo de tiempo y día con una duración de cirugías de 1 periodo de tiempo y la misma prioridad para todos los pacientes. De esta manera, el algoritmo deberá realizar la comparación con todos los elementos de la lista y además cumplirá con todas las condiciones de disponibilidad y compatibilidad

6.4. Experimentos

Los diferentes experimentos realizados para la parametrización y el análisis de desempeño del algoritmo GRASP y su implementación (Sección 6.4.1) se describen a continuación. La parametrización de α , Max_iter , β y γ se describe en la Sección 6.4.2. Además, se describen los experimentos realizados con el algoritmo GRASP para realizar la comparación con el modelo de programación lineal entero (Sección 6.5.1), instancias de mayor tamaño (Sección 6.5.2) y las medidas de desempeño utilizadas para evaluar la calidad de solución (Sección 6.6).

6.4.1 Implementación

El algoritmo GRASP es implementado en el lenguaje de programación *Python*, en el ambiente de programación *Spyder*. Se utilizó un procesador Intel Core i5 de 2.53GHz, con 4GB de memoria RAM, en la distribución de Linux: Ubuntu 12.04 de 64 bits.

6.4.2 Parametrización

En el algoritmo GRASP se hallan 4 parámetros, cuyos valores son establecidos por diferentes experimentos. El parámetro umbral $\alpha \in [0,1]$ el cuál determina el grado de aleatoriedad de la etapa constructiva al momento de seleccionar un elemento de la RCL (*Restricted Candidate List*); el parámetro Max_iter, el cual funciona como criterio de término para el algoritmo GRASP, es decir, cuantas veces se ejecutará todo el algoritmo para una instancia dada; y los parámetros β y γ , que son los ponderadores en la función *fitness* que poseen los mismos valores que los parámetros $\omega_1 = 0.7$ y $\omega_2 = 0.3$ respectivamente. Estos parámetros fueron establecidos en la implementación del modelo de programación lineal entero presentado en (Cartes, 2014), donde se ejecutó el modelo 9 veces en una misma instancia, evaluándose con respecto al valor de la función objetivo, cantidad de pacientes no programados y cantidad de periodos de sobretiempo utilizados en la programación.

6.4.3 Parámetro α

Para establecer el valor del parámetro α , se ejecutó el algoritmo 11 veces en una instancia dada (PCE 4, Anexo A.1), con un criterio de término Max_iter = 1, con distintos valores entre [0,1] como se observa en la Tabla 6.1. Estableciendo como el valor más adecuado de α aquel que entregara mejor calidad de resultado en términos de la función objetivo (columna 3), cantidad de pacientes no programados (columna 4) y cantidad de periodos de sobretiempo utilizados (columna 5) para cada valor de α establecido.

Con respecto a los mejores valores de la función objetivo se encuentran: $\alpha = 0.3$ y $\alpha = 0.8$. Por otro lado, la menor cantidad de pacientes no programados se encuentra en: $\alpha = 0.8$. Y finalmente, la menor cantidad de periodos de sobretiempo utilizados se encuentran en: $\alpha = 0.0$ y $\alpha = 0.3$. En resumen, con los resultados obtenidos para cada valor de α , los mejores valores de este parámetro son $\alpha = 0.3$ y $\alpha = 0.8$. Sin embargo, los periodos de sobretiempo utilizados para $\alpha = 0.8$ no son considerados como razonables (tres periodos de sobretiempo más en comparación con $\alpha = 0.3$ para una programación de dos días) para la instancia dada. Por esta razón, el parámetro α se establece para este algoritmo como 0.3.

TABLA 6.1. RESULTADOS DE LA PARAMETRIZACIÓN DE α

| Alpha α | Tiempo [seg] | Función Objetivo | #noprog | #st |
|----------------------------------|---------------------|-------------------------|----------------|------------|
| 0.0 | 0.16 | 5.316 | 2 | 3 |
| 0.1 | 0.15 | 5.097 | 2 | 6 |
| 0.2 | 0.15 | 4.710 | 3 | 7 |
| 0.3 | 0.26 | 5.384 | 2 | 4 |
| 0.4 | 0.20 | 5.206 | 2 | 5 |
| 0.5 | 0.19 | 5.162 | 2 | 6 |
| 0.6 | 0.21 | 5.051 | 2 | 8 |
| 0.7 | 0.30 | 5.054 | 2 | 7 |
| 0.8 | 0.16 | 5.348 | 1 | 7 |
| 0.9 | 0.19 | 5.247 | 2 | 8 |
| 1.0 | 0.17 | 5.148 | 3 | 8 |

6.4.4 Parámetro Max_iter

El valor del parámetro Max_iter se determinó ejecutando el algoritmo 8 veces en una instancia dada (PCE 4), donde el parámetro toma valores de 10 a 1000 para $\alpha = 0.3$. Estableciendo el valor más adecuado de Max_iter, como aquel que entregará mejor calidad de resultado en términos de la función objetivo.

Para el parámetro Max_iter se establece su valor elegido arbitrariamente, observando los mejores valores (véase Tabla 6.2) en términos del valor de la función objetivo (columna 2) y el tiempo de ejecución (columna 3).

Con respecto al valor de la función objetivo, los mejores valores para Max_iter se encuentran en: Max_iter = 1000 y Max_iter = 50. Por otro lado, los mejores valores para Max_iter con respecto al tiempo de ejecución se encuentran obviamente en los primeros valores para Max_iter. Cabe destacar, que esta instancia utilizada en la parametrización es pequeña y el tiempo de ejecución se considera importante al momento de ejecutar el algoritmo en instancias de mayor tamaño, es así como se puede distinguir que el tiempo de ejecución de Max_iter = 50

es más de 20 veces menor que en $\text{Max_iter} = 1000$. De esta manera, se establece $\text{Max_iter} = 50$, por calidad del valor de función objetivo y un tiempo razonable de ejecución para la instancia.

6.5. Generación de instancias

En esta sección se describen cómo se obtuvieron las instancias mostradas en este estudio. En primer lugar, se utilizaron las instancias ejecutadas en el trabajo anterior para comparar los desempeños del modelo de programación lineal entero y el algoritmo GRASP. En segundo lugar, ya que el modelo de programación lineal entero es ineficiente en instancias de mayor tamaño que las mostradas en el estudio anterior, en este estudio se muestran instancias de mayor tamaño para conocer cuál es el comportamiento del algoritmo en dichos escenarios.

TABLA 6.2. RESULTADOS DE LA PARAMETRIZACIÓN DE MAX_ITER

| Max_iter | Fitness | Tiempo [seg] |
|-----------|--------------|--------------|
| 10 | 5.316 | 0.84 |
| 20 | 5.364 | 1.40 |
| 30 | 5.393 | 2.13 |
| 50 | 5.459 | 3.50 |
| 100 | 5.378 | 6.68 |
| 200 | 5.449 | 13.63 |
| 500 | 5.767 | 36.07 |
| 1000 | 5.903 | 71.61 |

6.5.1 Instancias de comparación con el modelo de programación lineal entero

Las instancias de comparación con el modelo de programación lineal entero presentado en el capítulo 5 son elegidas para mostrar las diferencias en tiempo de ejecución y calidad de resultados en términos de las medidas de desempeño mostradas en la sección 6.6. Estas instancias se encuentran presentes en el trabajo de memoria de título (Cartes, 2014), las cuales pertenecen a las instancias PCE1-PCE10 y PCE17 (Anexo A.2).

6.5.2 Instancias de mayor tamaño para el algoritmo GRASP

Para generar las instancias de mayor tamaño se utilizaron los mismos datos obtenidos para el trabajo desarrollado anteriormente (Cartes, 2014), datos históricos de un hospital público, que posee 10 salas quirúrgicas en la Unidad de Pabellón de los cuales 2 son de uso exclusivo para el Servicio de Emergencias y 1 sala es de uso exclusivo de Cardiocirugía. En estas salas quirúrgicas los pacientes no son considerados electivos, por lo que se disponía de la información de 7 salas quirúrgicas restantes de la Unidad de Pabellón que poseen especialidades de: Otorrinolaringología, Oftalmología, Traumatología, Cardiología; y el resto de las salas quirúrgicas son consideradas como “libre” en donde se realizan el resto de las cirugías compatibles con dichas salas. Se disponía de la información de 15 médicos, con especialidades de: Otorrinolaringología, Traumatología, Cirugía Plástica, Cardiología y Cirugía General. La lista de espera de pacientes se generó de acuerdo a las tablas operatorias de la segunda y tercera semana de Enero de 2014, estableciendo dos instancias con 100 y 150 (IR100_15M e IR150_15M) pacientes en la lista de espera, con la compatibilidad existente de acuerdo a los médicos elegidos, además, la prioridad de cada paciente se estableció con números aleatorios entre $[0,1]$, simulando lo que sería la determinación de la prioridad del paciente por parte del personal clínico especializado. Se utilizó una jornada laboral diaria continuada de 8:00 a 16:00 horas, así, se utilizaron periodos de tiempo de 30 minutos y 2 horas permitidas para el sobretiempo. Por lo tanto, se consideraron 20 periodos de tiempo por día de programación. Para los tiempos de cirugía, se utilizaron tiempos estimados por personal clínico, incluyendo en éstos, los tiempos de preparación y limpieza de la sala quirúrgica. Los costos de utilización de las salas

quirúrgicas fueron costos estimados por personal clínico del servicio. Además, de las instancias IR100_15M e IR150_15M obtenidas con los datos históricos de las dos últimas semanas del mes de Enero del hospital público, se crearon 8 instancias IR150_17M hasta IR250_22M aumentando la cantidad de médicos hasta 22 médicos y la cantidad de pacientes hasta 250, manteniendo el mismo número de salas quirúrgicas y especialidades (Anexo A.3).

6.6. Medidas de Desempeño

Para conocer la calidad de solución entregada por el algoritmo GRASP, se usaron dos medidas de desempeño encontradas en la literatura y utilizadas en el trabajo anterior (Cartes, 2014). Estas medidas de desempeño son elegidas por la representación de lo buscado en el algoritmo GRASP, en términos de disminuir las listas de esperas por cirugías electivas y mejorar el uso de manera eficiente de los recursos en el servicio de pabellón. Por otro lado, estas medidas de desempeño permiten la comparación con el modelo de programación lineal entero.

6.6.1 Pacientes no programados por periodo de programación

Esta medida de desempeño es la más simple y no requiere mayor análisis. En esta medida, como su nombre lo dice, se contabilizan los pacientes que no son asignados en la programación entregada por el algoritmo.

6.6.2 Utilización de salas quirúrgicas

El porcentaje de utilización de salas quirúrgicas es calculado de acuerdo a lo propuesto por el Ministerio de Salud de Chile (Departamento de Estadísticas e Información de Salud, 2016) que se representa en la ecuación (6.2).

$$\text{Porcentaje de Utilización} = \frac{h_u + h_p}{h_d} \times 100 \quad (6.2)$$

Donde h_u son las horas totales mensuales utilizadas, h_p las horas mensuales de preparación (incluidas dentro de la estimación de tiempos de cirugía) y h_d el total de horas mensuales disponibles o semanales en lo que se refieren los resultados de este estudio.

Capítulo 7. Resultados y Análisis

7.1. Introducción

En este capítulo, se muestran los resultados obtenidos por el algoritmo GRASP en comparación con los obtenidos por el modelo de programación lineal entero en el trabajo anterior (Cartes, 2014). Además, se muestra y analiza el desempeño del algoritmo GRASP para instancias de mayor tamaño.

En la sección 7.2 se muestra, la comparación del algoritmo GRASP con el modelo de programación lineal entero en las instancias PCE1-PCE10.

En la sección 7.3 se detalla la comparación de las tablas operatorias obtenidas con el algoritmo GRASP con el modelo de programación lineal entero en la instancia PCE17.

Finalmente en la sección 7.4 se muestran los resultados obtenidos utilizando el algoritmo GRASP en instancias de mayor tamaño (IR100_15M-IR250_22M).

7.2. Comparación con el modelo de programación lineal entero en instancias ajustadas a la realidad

En la Tabla 7.1 se observan las características de las instancias ajustadas a la realidad (PCE1- PCE10), comparadas con el algoritmo GRASP, donde se detalla: el nombre de la instancia (columna 1), número de variables binarias (columna 2), número de variables enteras (columna 3), número de restricciones (columna 4)

Los resultados obtenidos en las instancias PCE1-PCE10 son mostrados en la Tabla 7.2, en donde las columnas correspondientes, al modelo programación lineal entero se detallan de la siguiente manera: Nombre de la instancia (columna 1), *GAP* entre la solución del problema relajado con la solución entera (columna 2), valor de la función objetivo (columna 3), solución óptima del problema relajado (columna 4), cantidad de pacientes no programados (columna 5), cantidad de periodos de sobretiempo utilizados en la programación (columna 6) y el tiempo de ejecución (columna 7). Además, para efectos de comparación del algoritmo con un solver más

eficiente, se realizó la comparación en el tiempo de ejecución con CPLEX (columna 8). Con respecto a los resultados del algoritmo GRASP, para cada instancia se detallan: el valor de la función *fitness* (columna 9), cantidad de pacientes no programados (columna 10), cantidad de periodos de sobretiempo utilizados en la programación (columna 11) y el tiempo de ejecución (columna 12).

En los resultados se observa, que en las instancias PCE1 y PCE2 los dos métodos (IP y GRASP) entregan el resultado óptimo y en tiempos similares. En las instancias PCE3, PCE4, PCE5 y PCE6, solo el modelo IP entregó el resultado óptimo, sin embargo el algoritmo GRASP entrega resultados muy cercanos al óptimo, sobre todo en términos de cantidad de pacientes no programados y cantidad de periodos de sobretiempo utilizados. Por otro lado, en las instancias PCE9 y PCE10, mientras el modelo IP resuelto con el solver de GLPK, no encuentra solución durante un periodo de ejecución de 4 horas, el algoritmo GRASP entrega la solución óptima, en un tiempo de ejecución de 8.13 y 5.78 segundos respectivamente y el modelo resuelto con CPLEX posee tiempos de ejecución de 0.58 y 0.43.

A pesar de que los tiempos de resolución más cortos se encuentran resolviendo el problema con CPLEX, no se realizó un estudio más profundo con esta herramienta, ya que para la realidad de hospital público chileno se hace difícil costear una licencia de este tipo.

TABLA 7.1. CARACTERÍSTICAS DE LAS INSTANCIAS AJUSTADAS A LA REALIDAD

| Nombre de la instancia | # vars binarias | # vars enteras | # restricciones |
|------------------------|-----------------|----------------|-----------------|
| PCE1 | 365 | 10 | 95 |
| PCE2 | 710 | 20 | 915 |
| PCE3 | 1055 | 30 | 1325 |
| PCE4 | 1975 | 30 | 2390 |
| PCE5 | 3855 | 30 | 3141 |
| PCE6 | 4770 | 30 | 3492 |
| PCE7 | 5120 | 30 | 3541 |
| PCE8 | 5700 | 30 | 5260 |
| PCE9 | 7515 | 30 | 5931 |
| PCE10 | 5685 | 30 | 6231 |

TABLA 7.2. RESULTADOS DE INSTANCIAS DE COMPARACIÓN

| Inst | Gap (%) | Obj | IP | | # st | GPLK T(seg) | CPLEX T (seg) | GRASP | | | |
|-------|---------|--------------|--------------|------|------|----------------|------------------|--------------|------|-----|--------|
| | | | LP | #nop | | | | Fitness | #nop | #st | T(seg) |
| PCE1 | 0.00 | 1.183 | 1.183 | 1 | 0 | 0.20 | 0.13 | 1.183 | 1 | 0 | 0.14 |
| PCE2 | 0.00 | 2.863 | 2.863 | 1 | 0 | 0.20 | 0.06 | 2.863 | 1 | 0 | 0.36 |
| PCE3 | 0.00 | 5.117 | 5.117 | 4 | 0 | 1.01 | 0.04 | 4.941 | 4 | 1 | 0.53 |
| PCE4 | 0.00 | 5.789 | 5.789 | 2 | 4 | 3.67 | 0.10 | 5.384 | 2 | 4 | 3.50 |
| PCE5 | 0.00 | 4.039 | 4.039 | 1 | 0 | 375.10 | 0.14 | 4.018 | 1 | 0 | 2.41 |
| PCE6 | 0.00 | 6.833 | 6.833 | 0 | 0 | 680.20 | 0.20 | 6.781 | 0 | 1 | 2.69 |
| PCE7 | 0.60 | 5.746 | 5.782 | 0 | 2 | 10.35 | 0.10 | 5.764 | 0 | 1 | 3.70 |
| PCE8 | 1.80 | 7.088 | 7.217 | 0 | 1 | 14400.00 | 0.85 | 6.914 | 0 | 3 | 5.20 |
| PCE9 | - | - | 6.020 | - | - | 14400.00 | 0.43 | 6.020 | 0 | 0 | 8.13 |
| PCE10 | - | - | 5.236 | - | - | 14400.00 | 0.32 | 5.236 | 1 | 0 | 5.78 |

7.3. Comparación con el modelo IP en instancia con datos reales

Ya que de las instancias ejecutadas en el trabajo anterior (Cartes, 2014), la única instancia en entregar resultados es la instancia PCE17, ésta es ejecutada y comparada con los resultados entregados por el algoritmo GRASP.

En la Tabla 7.3 se muestra el resumen de la comparación de resultados en la instancia PCE17. Como se puede observar existe una gran diferencia en el tiempo de ejecución de los dos métodos, por un lado modelo IP resuelto con el MIP solver de GLPK entregó resultados en 130800 segundos (columna 5) en comparación 0.25 segundos con el MIP solver de CPLEX y con 1.99 segundos del algoritmo GRASP.

En términos del valor de la función objetivo (columna 2) y la función *fitness* (columna 6) existe una diferencia de 0.398 a favor del modelo IP, esto debido al uso de periodos de sobretiempo programados con el algoritmo GRASP (columna 9), ya que en la cantidad de pacientes no programados (columna 3 y columna 8) el modelo IP y el algoritmo GRASP programaron todos los pacientes de la instancia.

**TABLA 7.3.RESULTADOS OBTENIDOS EN INSTANCIA DE COMPARACIÓN DE LOS MÉTODOS
CON DATOS REALES**

| Instancia | Obj | IP | | GLPK Tiempo ejecución (seg) | CPLEX Tiempo ejecución (seg) | GRASP | | | |
|-----------|-------|-------------|-----|--------------------------------------|---------------------------------------|---------|---------|-----|------------------------------|
| | | # noprog | #st | | | Fitness | #noprog | #st | Tiempo ejecución (seg) |
| PCE17 | 3.533 | 0 | 1 | 130800 | 0.19 | 3.137 | 0 | 4 | 1.99 |

7.3.1 Tablas operatorias

Una manera más gráfica de comparar los resultados del modelo IP y el algoritmo GRASP, es por medio de las tablas operatorias, mostradas en las tablas 7.4, 7.5 y 7.6 correspondientes a las tablas operatorias de las salas quirúrgicas 1 (Otorrinolaringología), 2 (Traumatología) y 3 (Libre) respectivamente. En cada tabla descrita anteriormente, se muestra la tabla operatoria entregada por el modelo IP al lado izquierdo y algoritmo GRASP al lado derecho.

En la Tabla 7.4, está la programación correspondiente a la sala 1, con la especialidad de otorrinolaringología se observa que la utilización de los periodos de tiempo son exactamente iguales, utilizando la disponibilidad de tiempo completa del médico 5, además utilizando un periodo de sobretiempo en cada caso (en color rojo se muestra el bloque de sobretiempo utilizado, correspondiente a 30 minutos). La única diferencia en esta tabla operatoria, es el orden de la programación de los pacientes.

TABLA 7.4. TABLA OPERATORIA PARA LA SALA 1, EXCLUSIVA DE LA ESPECIALIDAD DE OTORRINOLARINGOLOGÍA

| t | IP | | GRASP | |
|----|----------------------|-----------------------|----------------------|----------------------|
| | Lunes | Martes | Lunes | Martes |
| 1 | Paciente 14/Médico 5 | | Paciente 13/Médico 5 | |
| 2 | Paciente 14/Médico 5 | | Paciente 13/Médico 5 | |
| 3 | Paciente 14/Médico 5 | | Paciente 13/Médico 5 | |
| 4 | Paciente 14/Médico 5 | | Paciente 13/Médico 5 | |
| 5 | Paciente 14/Médico 5 | | Paciente 13/Médico 5 | |
| 6 | Paciente 13/Médico 5 | | Paciente 13/Médico 5 | |
| 7 | Paciente 13/Médico 5 | | Paciente 13/Médico 5 | |
| 8 | Paciente 13/Médico 5 | | Paciente 15/Médico 5 | |
| 9 | Paciente 13/Médico 5 | | Paciente 15/Médico 5 | |
| 10 | Paciente 13/Médico 5 | | Paciente 15/Médico 5 | |
| 11 | Paciente 13/Médico 5 | | Paciente 15/Médico 5 | |
| 12 | Paciente 13/Médico 5 | | Paciente 15/Médico 5 | |
| 13 | | Paciente 15/ Médico 5 | | Paciente 14/Médico 5 |
| 14 | | Paciente 15/ Médico 5 | | Paciente 14/Médico 5 |
| 15 | | Paciente 15/ Médico 5 | | Paciente 14/Médico 5 |
| 16 | | Paciente 15/ Médico 5 | | Paciente 14/Médico 5 |
| 17 | | Paciente 15/ Médico 5 | | Paciente 14/Médico 5 |
| 18 | | | | |
| 19 | | | | |
| 20 | | | | |

La Tabla 7.5, correspondiente a la tabla operatoria de la sala 2, con la especialidad de traumatología se observa una similitud con la cantidad de periodos programados por día entre el modelo IP y el algoritmo GRASP. Se destaca una mayor concentración en los periodos programados el día lunes, en cambio, para el día martes se programa solo una cirugía en los dos casos.

En la programación de las cirugías de esta sala, ningún método utiliza periodos de sobretiempo en la programación ya que los periodos 17-20 no se programan pacientes.

TABLA 7.5. TABLA OPERATORIA PARA LA SALA 2, EXCLUSIVA DE LA ESPECIALIDAD TRAUMATOLOGÍA.

| t | IP | | GRASP | |
|----|----------------------|---------------------|----------------------|---------------------|
| | Lunes | Martes | Lunes | Martes |
| 1 | Paciente 9/Médico 4 | | Paciente 10/Médico 4 | |
| 2 | Paciente 9/Médico 4 | | Paciente 10/Médico 4 | |
| 3 | Paciente 9/Médico 4 | | Paciente 10/Médico 4 | |
| 4 | Paciente 9/Médico 4 | | Paciente 10/Médico 4 | |
| 5 | Paciente 9/Médico 4 | Paciente 8/Médico 4 | Paciente 10/Médico 4 | |
| 6 | Paciente 9/Médico 4 | Paciente 8/Médico 4 | Paciente 10/Médico 4 | |
| 7 | Paciente 10/Médico 4 | Paciente 8/Médico 4 | | |
| 8 | Paciente 10/Médico 4 | Paciente 8/Médico 4 | | |
| 9 | Paciente 10/Médico 4 | Paciente 8/Médico 4 | Paciente 8/Médico 4 | |
| 10 | Paciente 10/Médico 4 | | Paciente 8/Médico 4 | Paciente 9/Médico 4 |
| 11 | Paciente 10/Médico 4 | | Paciente 8/Médico 4 | Paciente 9/Médico 4 |
| 12 | Paciente 10/Médico 4 | | Paciente 8/Médico 4 | Paciente 9/Médico 4 |
| 13 | | | Paciente 8/Médico 4 | Paciente 9/Médico 4 |
| 14 | | | | Paciente 9/Médico 4 |
| 15 | | | | Paciente 9/Médico 4 |
| 16 | | | | |
| 17 | | | | |
| 18 | | | | |
| 19 | | | | |
| 20 | | | | |

Para el caso de la Tabla 7.6, la cual muestra la tabla operatoria perteneciente a la sala 3, clasificada como libre se observa una programación de cirugías más variada entre los dos métodos en comparación con las otras dos salas. Además de notar la diferencia en el orden de la programación de cirugías, se destaca en el algoritmo GRASP el uso de 3 periodos de sobretiempo en la programación del día martes (marcado con rojo), siendo el equivalente a 1,5 horas de sobretiempo.

TABLA 7.6. TABLA OPERATORIA PARA LA SALA 3, CONSIDERADA COMO LIBRE.

| t | IP | | GRASP | |
|----|----------------------|----------------------|----------------------|----------------------|
| | Lunes | Martes | Lunes | Martes |
| 1 | | Paciente 11/Médico 3 | | Paciente 12/Médico 3 |
| 2 | Paciente 4/Médico 2 | Paciente 11/Médico 3 | Paciente 5/Médico 2 | Paciente 12/Médico 3 |
| 3 | Paciente 4/Médico 2 | Paciente 11/Médico 3 | Paciente 5/Médico 2 | Paciente 12/Médico 3 |
| 4 | Paciente 4/Médico 2 | Paciente 3/Médico 1 | Paciente 5/Médico 2 | Paciente 1/Médico 1 |
| 5 | Paciente 6/Médico 2 | Paciente 3/Médico 1 | Paciente 4/Médico 2 | Paciente 1/Médico 1 |
| 6 | Paciente 6/Médico 2 | Paciente 3/Médico 1 | Paciente 4/Médico 2 | Paciente 1/Médico 1 |
| 7 | Paciente 6/Médico 2 | Paciente 7/Médico 2 | Paciente 4/Médico 2 | Paciente 6/Médico 2 |
| 8 | Paciente 5/Médico 2 | Paciente 7/Médico 2 | Paciente 11/Médico 3 | Paciente 6/Médico 2 |
| 9 | Paciente 5/Médico 2 | Paciente 7/Médico 2 | Paciente 11/Médico 3 | Paciente 6/Médico 2 |
| 10 | Paciente 5/Médico 2 | Paciente 2/Médico 1 | Paciente 11/Médico 3 | Paciente 3/Médico 1 |
| 11 | | Paciente 2/Médico 1 | | Paciente 3/Médico 1 |
| 12 | | | | Paciente 3/Médico 1 |
| 13 | | Paciente 1/Médico 1 | | Paciente 2/Médico 1 |
| 14 | Paciente 12/Médico 3 | Paciente 1/Médico 1 | | Paciente 2/Médico 1 |
| 15 | Paciente 12/Médico 3 | Paciente 1/Médico 1 | | |
| 16 | Paciente 12/Médico 3 | | | |
| 17 | | | Paciente 7/Médico 2 | |
| 18 | | | Paciente 7/Médico 2 | |
| 19 | | | Paciente 7/Médico 2 | |
| 20 | | | | |

7.3.2 Utilización de las salas quirúrgicas

La utilización de las salas quirúrgica para la instancia PCE17 se muestra en la Tabla 7.7, en la cual se observa una utilización de salas muy similar a la obtenida por el modelo IP, debido a la programación entregada por los dos métodos. También, se observa una utilización mensual proyectada exactamente igual en los dos métodos, con la excepción de la sala quirúrgica tres, en donde existe una menor utilización con el algoritmo GRASP, ya que éste utiliza periodos de sobretiempo en la programación entregada en dicha sala.

TABLA 7.7. UTILIZACIÓN DE LAS SALAS QUIRÚRGICAS INSTANCIA PCE17

| Sala | IP | | | GRASP | | |
|------|-------------------------|--------------------------|----------------------------------|-------------------------|--------------------------|----------------------------------|
| | % Utilización día Lunes | % Utilización día Martes | % Utilización mensual proyectado | % Utilización día Lunes | % Utilización día Martes | % Utilización mensual proyectado |
| 1 | 75.00 | 25.00 | 50.00 | 75.00 | 25.00 | 50.00 |
| 2 | 75.00 | 31.25 | 53.13 | 68.75 | 37.50 | 53.13 |
| 3 | 75.00 | 87.50 | 81.25 | 56.25 | 87.50 | 71.88 |

7.4. Instancias de mayor tamaño

Los resultados entregados por las instancias IR100_15M hasta IR250_22M son mostrados en el cuadro resumen (Tabla 7.8) que destaca: el valor de la función *fitness* (columna 2), la cantidad de pacientes no programados (columna 3), la cantidad de periodos de sobretiempo utilizado en la programación (columna 4), el tiempo de ejecución del algoritmo (columna 5) y el orden de complejidad O del algoritmo para cada instancia (columna 6). Para el caso de las instancias IR100_15M e IR150_15M se muestran las tablas operatorias generadas para cada día de la semana en las secciones 7.4.1 y 7.4.2 respectivamente. Además, en la sección 7.4.3 se presenta la tabla con el porcentaje de utilización de las salas quirúrgicas.

A medida que el orden de complejidad computacional aumenta, el tiempo de ejecución también lo hace. Así, es posible observar de mejor manera cómo se comporta el algoritmo con respecto a su complejidad computacional en las instancias IR175_17M, IR175_18M e IR175_20M que muestran el cambio en el tiempo de ejecución al aumentar solo una variable, en este caso, la cantidad de médicos. Entre la instancia IR175_17M e IR175_18M, aumenta en uno la cantidad de médicos lo que equivale a un 5,88% en la complejidad O y el tiempo de ejecución del algoritmo aumenta en un 25,28%, además para el caso entre la instancia IR175_18M y IR175_20M que aumenta la cantidad de médicos en 2, siendo un aumento del 11,11% en la complejidad O y el tiempo de ejecución del algoritmo GRASP aumenta en un 33,62% observándose el comportamiento no lineal del algoritmo.

TABLA 7.8. RESUMEN RESULTADOS OBTENIDOS EN INSTANCIAS IR100_15M HASTA IR250_22M

| Instancia | Fitness | #noprogramado | #st | Tiempo ejecución (seg) | $O(Max_iter \times p \times m \times s \times t \times i)$ |
|------------------|----------------|----------------------|------------|-------------------------------|---|
| IR100_15M | 33.295 | 10 | 79 | 382.62 | $O(52000000)$ |
| IR150_15M | 42.413 | 35 | 85 | 594.52 | $O(78750000)$ |
| IR150_17M | 42.300 | 29 | 92 | 1018.35 | $O(89250000)$ |
| IR175_17M | 47.575 | 42 | 103 | 1044.65 | $O(104125000)$ |
| IR175_18M | 47.708 | 41 | 103 | 1308.78 | $O(110250000)$ |
| IR175_20M | 48.208 | 38 | 105 | 1748.77 | $O(122500000)$ |
| IR200_20M | 56.551 | 41 | 112 | 2344.55 | $O(140000000)$ |
| IR225_20M | 60.358 | 59 | 111 | 3316.26 | $O(157500000)$ |
| IR225_22M | 60.339 | 61 | 110 | 3715.69 | $O(173250000)$ |
| IR250_22M | 64.703 | 76 | 120 | 4728.88 | $O(192500000)$ |

7.4.1 Tablas operatorias instancia IR100_15M

A continuación se muestran las tablas operatorias de la instancia IR100_15M para los días: lunes (Tabla 7.9), martes (Tabla 7.10), miércoles (Tabla 7.11), jueves (Tabla 7.12) y viernes (Tabla 7.13). Donde se programan en los 20 periodos de tiempo en cada jornada laboral y cada sala quirúrgica, el paciente y el médico a cargo de la cirugía.

La programación de cirugías entregada por el algoritmo muestra una mayor cantidad de cirugías programadas durante los primeros días de la semana. Esto quiere decir que el algoritmo va programando desde lo relativamente más temprano dentro del día y dentro de la semana, ya que depende directamente del parámetro α . Además, se observa una utilización de salas quirúrgicas en periodos de sobretiempos que disminuye en los últimos días. Sin embargo, la cantidad de pacientes no programados es mínima en comparación con el tamaño de la instancia y la utilización de la disponibilidad de tiempo de los médicos es alta, dejando ventanas de tiempo disponibles para futuras cirugías en los días jueves y viernes.

TABLA 7.9. TABLA OPERATORIA IR100_15M PARA EL DÍA LUNES

| t | Lunes | | | | | | |
|----|---------|---------|---------|--------|--------|--------|---------|
| | Sala 1 | Sala 2 | Sala 3 | Sala 4 | Sala 5 | Sala 6 | Sala 7 |
| 1 | P78/M11 | P88/M13 | | P30/M3 | | P64/M4 | |
| 2 | P78/M11 | P88/M13 | | P30/M3 | P44/M6 | P64/M4 | |
| 3 | | P88/M13 | | P30/M3 | P44/M6 | P64/M4 | |
| 4 | P62/M11 | P88/M13 | P39/M12 | | | | |
| 5 | P62/M11 | P88/M13 | P39/M12 | | P66/M8 | P10/M6 | |
| 6 | P89/M11 | P84/M13 | P39/M12 | P54/M7 | P66/M8 | P10/M6 | |
| 7 | P89/M11 | P84/M13 | P39/M12 | P54/M7 | P18/M4 | P10/M6 | |
| 8 | P89/M11 | P84/M13 | P39/M12 | P54/M7 | P18/M4 | | |
| 9 | | P84/M13 | P39/M12 | P83/M9 | P18/M4 | P12/M7 | |
| 10 | | P84/M13 | | P83/M9 | | P12/M7 | |
| 11 | P94/M10 | | | P83/M9 | | | P79/M15 |
| 12 | P94/M10 | | | | | | P79/M15 |
| 13 | P94/M10 | | | | P76/M4 | | P79/M15 |
| 14 | P94/M10 | | | | P76/M4 | | P79/M15 |
| 15 | P94/M10 | | | | P76/M4 | | P79/M15 |
| 16 | P47/M10 | | | | | P59/M4 | P79/M15 |
| 17 | P47/M10 | P90/M14 | | | P4/M6 | P59/M4 | |
| 18 | P47/M10 | P90/M14 | | P5/M2 | P4/M6 | P59/M4 | |
| 19 | P47/M10 | P90/M14 | | P5/M2 | P4/M6 | | |
| 20 | P47/M10 | P90/M14 | | | | | |

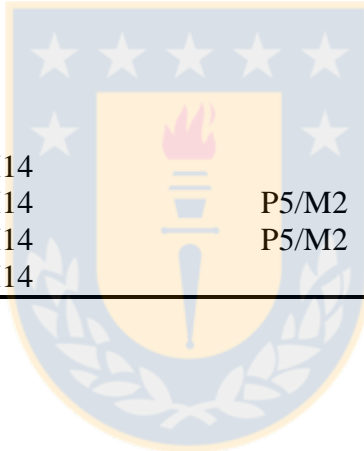


TABLA 7.10. TABLA OPERATORIA IR100_15M PARA EL DÍA MARTES

| t | Martes | | | | | | |
|----|---------|---------|---------|--------|--------|--------|---------|
| | Sala 1 | Sala 2 | Sala 3 | Sala 4 | Sala 5 | Sala 6 | Sala 7 |
| 1 | | | P38/M12 | | P9/M6 | | |
| 2 | P67/M10 | P91/M13 | P38/M12 | | P9/M6 | P4/M2 | |
| 3 | P67/M10 | P91/M13 | P38/M12 | P19/M4 | P9/M6 | P4/M2 | |
| 4 | P67/M10 | P91/M13 | P38/M12 | P19/M4 | | P4/M2 | |
| 5 | P67/M10 | P91/M13 | P38/M12 | P19/M4 | P32/M2 | | |
| 6 | P67/M10 | P91/M13 | P26/M12 | P11/M6 | P32/M2 | P16/M3 | |
| 7 | | | P26/M12 | P11/M6 | P75/M2 | P16/M3 | |
| 8 | | P82/M13 | P26/M12 | P11/M6 | P75/M2 | P16/M3 | |
| 9 | P93/M11 | P82/M13 | P26/M12 | | | | |
| 10 | P93/M11 | P82/M13 | P26/M12 | P40/M6 | P92/M1 | | |
| 11 | | P82/M13 | P26/M12 | P40/M6 | P92/M1 | | |
| 12 | | | P61/M14 | P40/M6 | P36/M5 | | P57/M15 |
| 13 | | P58/M13 | P61/M14 | P1/M1 | P36/M5 | P43/M6 | P57/M15 |
| 14 | | P58/M13 | P61/M14 | P1/M1 | P36/M5 | P43/M6 | P57/M15 |
| 15 | | P58/M13 | P61/M14 | | P8/M5 | P42/M6 | P57/M15 |
| 16 | | P58/M13 | P61/M14 | | P8/M5 | P42/M6 | P57/M15 |
| 17 | P21/M10 | P58/M13 | | | P8/M5 | P42/M6 | P57/M15 |
| 18 | P21/M10 | | | P97/M4 | P3/M1 | P6/M5 | |
| 19 | P21/M10 | P53/M12 | | P97/M4 | P3/M1 | P6/M5 | |
| 20 | P21/M10 | P53/M12 | | P97/M4 | | P6/M5 | |

TABLA 7.11. TABLA OPERATORIA IR100_15M PARA EL DÍA MIÉRCOLES

| t | Miércoles | | | | | | |
|----|-----------|---------|---------|--------|--------|--------|---------|
| | Sala 1 | Sala 2 | Sala 3 | Sala 4 | Sala 5 | Sala 6 | Sala 7 |
| 1 | | | | | P28/M3 | | |
| 2 | | P85/M14 | | P17/M4 | P28/M3 | | |
| 3 | | P85/M14 | | P17/M4 | P28/M3 | | P98/M15 |
| 4 | | P85/M14 | | P17/M4 | | | P98/M15 |
| 5 | | P65/M13 | | | | | P98/M15 |
| 6 | | P65/M13 | | P7/M5 | | | P98/M15 |
| 7 | | P65/M13 | | P7/M5 | P71/M2 | P29/M3 | P98/M15 |
| 8 | | P65/M13 | | P7/M5 | P71/M2 | P29/M3 | P98/M15 |
| 9 | | P65/M13 | | | P71/M2 | P29/M3 | P60/M15 |
| 10 | | | | | P71/M2 | | P60/M15 |
| 11 | | | P95/M13 | | P15/M3 | P35/M5 | P60/M15 |
| 12 | | | P95/M13 | | P15/M3 | P35/M5 | P60/M15 |
| 13 | | | P95/M13 | | P15/M3 | P35/M5 | P60/M15 |
| 14 | | | P95/M13 | | | | P60/M15 |
| 15 | | | P95/M13 | | | | P60/M15 |
| 16 | | P68/M14 | | | | P45/M5 | |
| 17 | P48/M10 | P68/M14 | | | | P45/M5 | |
| 18 | P48/M10 | P68/M14 | | P81/M8 | | P45/M5 | |
| 19 | P48/M10 | P68/M14 | | P81/M8 | P1/M1 | P45/M5 | |
| 20 | P48/M10 | P68/M14 | | | P1/M1 | P45/M5 | |

TABLA 7.12. TABLA OPERATORIA IR100_15M PARA EL DÍA JUEVES

| t | Jueves | | | | | | |
|----|---------|--------|---------|--------|--------|--------|--------|
| | Sala 1 | Sala 2 | Sala 3 | Sala 4 | Sala 5 | Sala 6 | Sala 7 |
| 1 | | | | | | | |
| 2 | P87/M11 | | | | | | |
| 3 | P87/M11 | | P96/M14 | | P33/M1 | | |
| 4 | | | P96/M14 | | P33/M1 | P13/M7 | |
| 5 | | | P96/M14 | | P33/M1 | P13/M7 | |
| 6 | | | P96/M14 | | | P13/M7 | |
| 7 | | | P96/M14 | | | | |
| 8 | | | | | | | |
| 9 | P52/M11 | | | | | | |
| 10 | P52/M11 | | | | P55/M8 | | |
| 11 | P20/M10 | | | | P55/M8 | P69/M9 | |
| 12 | P20/M10 | | | | P55/M8 | P69/M9 | |
| 13 | P20/M10 | | | P34/M5 | P51/M6 | P69/M9 | |
| 14 | P20/M10 | | | P34/M5 | P51/M6 | | |
| 15 | | | | P34/M5 | P51/M6 | | |
| 16 | P80/M10 | | | | | | |
| 17 | P80/M10 | | | | | | |
| 18 | P80/M10 | | | P72/M6 | | | |
| 19 | P80/M10 | | | P72/M6 | | | |
| 20 | P80/M10 | | | P72/M6 | | | |

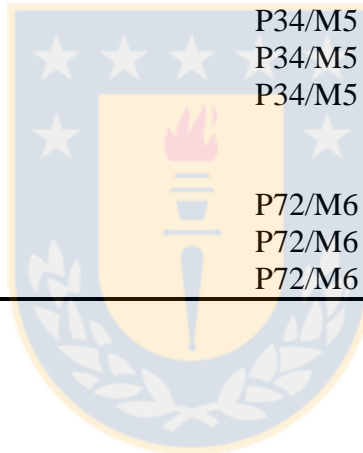


TABLA 7.13. TABLA OPERATORIA IR100_15M PARA EL DÍA VIERNES

| t | Viernes | | | | | | |
|----|---------|----------|---------|--------|--------|--------|---------|
| | Sala 1 | Sala 2 | Sala 3 | Sala 4 | Sala 5 | Sala 6 | Sala 7 |
| 1 | | | | | | | |
| 2 | P56/M10 | | P25/M12 | | | | P99/M15 |
| 3 | P56/M10 | | P25/M12 | | | | P99/M15 |
| 4 | P56/M10 | | P25/M12 | | | | P99/M15 |
| 5 | P56/M10 | | P25/M12 | | | | P99/M15 |
| 6 | P56/M10 | | P25/M12 | | | | P99/M15 |
| 7 | P22/M10 | | | | | | P99/M15 |
| 8 | P22/M10 | | | | | | P99/M15 |
| 9 | P22/M10 | | | | | | |
| 10 | P23/M10 | | | | | | P63/M15 |
| 11 | P23/M10 | | P37/M12 | | | | P63/M15 |
| 12 | P23/M10 | | P37/M12 | | | | P63/M15 |
| 13 | | | P37/M12 | | | | P63/M15 |
| 14 | | | P37/M12 | | | | P63/M15 |
| 15 | | P100/M13 | | | | | P63/M15 |
| 16 | P74/M10 | P100/M13 | | | | | |
| 17 | P74/M10 | P100/M13 | | | | | |
| 18 | P74/M10 | P100/M13 | | P14/M8 | | | |
| 19 | P74/M10 | P100/M13 | | P14/M8 | | | |
| 20 | | | | P14/M8 | | | |

7.4.2 Tablas operatorias instancia IR150_15M

En esta sección se muestran las tablas operatorias de la instancia IR150_15M para los días: lunes (Tabla 8.14), martes (Tabla 8.15), miércoles (Tabla 8.16), jueves (Tabla 8.17) y viernes (Tabla 8.18). Donde se programan en los 20 periodos de tiempo de cada jornada laboral y cada sala quirúrgica, el paciente y el médico a cargo de la cirugía.

Al igual que en la instancia IR100_15M la cantidad de pacientes programados es mayor en los primeros tres días de la semana (lunes, martes y miércoles) de acuerdo al funcionamiento del algoritmo. También se observa una mayor utilización de sobretiempo en las salas quirúrgicas en los primeros días de la semana, utilizando por ejemplo el día lunes 23 periodos de sobretiempo, en comparación con 8 periodos de sobretiempo el día viernes.

La menor utilización de las salas quirúrgicas se encuentra en las salas con mayor especificidad, es decir, para las salas catalogadas como libre (Salas 4,5 y 6) existe una mayor utilización ya que es compatible con un mayor número de cirugías y médicos, en comparación con la sala de cardiología (Sala 7) que es compatible con un tipo de cirugía y solo un médico para esta instancia (M15).

TABLA 7.14. TABLA OPERATORIA IR150_15M PARA EL DÍA LUNES

| t | Lunes | | | | | | |
|----|----------|---------|----------|---------|----------|---------|----------|
| | Sala 1 | Sala 2 | Sala 3 | Sala 4 | Sala 5 | Sala 6 | Sala 7 |
| 1 | | P82/M13 | | P30/M3 | P125/M9 | | |
| 2 | P121/M11 | P82/M13 | P25/M12 | P30/M3 | P125/M9 | P14/M8 | |
| 3 | P121/M11 | P82/M13 | P25/M12 | P30/M3 | P17/M4 | P14/M8 | |
| 4 | P89/M11 | P82/M13 | P25/M12 | | P17/M4 | P14/M8 | |
| 5 | P89/M11 | P65/M13 | P25/M12 | P66/M5 | P17/M4 | | |
| 6 | P89/M11 | P65/M13 | P25/M12 | P66/M5 | | P147/M8 | |
| 7 | P101/M11 | P65/M13 | P140/M12 | | P40/M6 | P147/M8 | |
| 8 | P101/M11 | P65/M13 | P140/M12 | P54/M7 | P40/M6 | P147/M8 | |
| 9 | | P65/M13 | P140/M12 | P54/M7 | P40/M6 | P81/M8 | |
| 10 | | | P140/M12 | P54/M7 | | P81/M8 | |
| 11 | P138/M10 | | P140/M12 | P112/M4 | P148/M8 | | |
| 12 | P138/M10 | | P140/M12 | P112/M4 | P148/M8 | | |
| 13 | P138/M10 | | | P112/M4 | P148/M8 | | |
| 14 | P138/M10 | | | | P64/M4 | | P150/M15 |
| 15 | P138/M10 | | | | P64/M4 | | P150/M15 |
| 16 | | | | | P64/M4 | | P150/M15 |
| 17 | P20/M10 | | P117/M14 | P10/M6 | | P71/M1 | P150/M15 |
| 18 | P20/M10 | P53/M12 | P117/M14 | P10/M6 | P103/M11 | P71/M1 | P150/M15 |
| 19 | P20/M10 | P53/M12 | P117/M14 | P10/M6 | P103/M11 | P71/M1 | P150/M15 |
| 20 | | | P117/M14 | | P103/M11 | P71/M1 | P150/M15 |

TABLA 7.15. TABLA OPERATORIA IR150_15M PARA EL DÍA MARTES

| t | Martes | | | | | | |
|----|----------|----------|---------|--------|---------|---------|---------|
| | Sala 1 | Sala 2 | Sala 3 | Sala 4 | Sala 5 | Sala 6 | Sala 7 |
| 1 | | P100/M14 | P39/M12 | P76/M3 | P51/M6 | P134/M8 | |
| 2 | P94/M10 | P100/M14 | P39/M12 | P76/M3 | P51/M6 | P134/M8 | |
| 3 | P94/M10 | P100/M14 | P39/M12 | P76/M3 | P51/M6 | P134/M8 | |
| 4 | P94/M10 | P100/M14 | P39/M12 | P6/M5 | P143/M2 | P11/M6 | |
| 5 | P94/M10 | P100/M14 | P39/M12 | P6/M5 | P143/M2 | P11/M6 | |
| 6 | P94/M10 | | P39/M12 | P6/M5 | | P11/M6 | |
| 7 | P87/M11 | | P37/M12 | | | | |
| 8 | P87/M11 | P88/M13 | P37/M12 | | P18/M4 | P34/M5 | |
| 9 | | P88/M13 | P37/M12 | P43/M6 | P18/M4 | P34/M5 | |
| 10 | P113/M11 | P88/M13 | P37/M12 | P43/M6 | P18/M4 | P34/M5 | |
| 11 | P113/M11 | P88/M13 | P37/M12 | P2/M1 | P45/M5 | P5/M2 | P99/M15 |
| 12 | | P88/M13 | | P2/M1 | P45/M5 | P5/M2 | P99/M15 |
| 13 | | | | P42/M6 | P45/M5 | | P99/M15 |
| 14 | | | | P42/M6 | P45/M5 | P33/M1 | P99/M15 |
| 15 | | | | P42/M6 | P45/M5 | P33/M1 | P99/M15 |
| 16 | | P91/M13 | P90/M14 | | P32/M2 | P33/M1 | P99/M15 |
| 17 | P139/M11 | P91/M13 | P90/M14 | P34/M5 | P32/M2 | | P99/M15 |
| 18 | P139/M11 | P91/M13 | P90/M14 | P34/M5 | | P116/M3 | |
| 19 | P139/M11 | P91/M13 | P90/M14 | P34/M5 | P3/M1 | P116/M3 | |
| 20 | P139/M11 | P91/M13 | | | P3/M1 | P116/M3 | |

TABLA 7.16. TABLA OPERATORIA IR150_15M PARA EL DÍA MIÉRCOLES

| t | Miércoles | | | | | | |
|----|-----------|----------|----------|--------|---------|---------|---------|
| | Sala 1 | Sala 2 | Sala 3 | Sala 4 | Sala 5 | Sala 6 | Sala 7 |
| 1 | | | | | P128/M2 | | |
| 2 | P96/M14 | | | P69/M9 | P128/M2 | | |
| 3 | P96/M14 | | | P69/M9 | P28/M3 | | |
| 4 | P96/M14 | | P107/M13 | P69/M9 | P28/M3 | | P57/M15 |
| 5 | P96/M14 | | P107/M13 | P8/M5 | P28/M3 | P83/M9 | P57/M15 |
| 6 | P96/M14 | | P107/M13 | P8/M5 | P19/M4 | P83/M9 | P57/M15 |
| 7 | | | P107/M13 | P8/M5 | P19/M4 | P83/M9 | P57/M15 |
| 8 | | | P107/M13 | P92/M2 | P19/M4 | | P57/M15 |
| 9 | | P130/M14 | P107/M13 | P92/M2 | P29/M3 | P115/M5 | P57/M15 |
| 10 | | P130/M14 | | | P29/M3 | P115/M5 | |
| 11 | | P130/M14 | | | P29/M3 | P115/M5 | |
| 12 | | P130/M14 | | | P122/M2 | P16/M3 | |
| 13 | | P130/M14 | | | P122/M2 | P16/M3 | |
| 14 | | P95/M13 | P68/M14 | | | P16/M3 | |
| 15 | | P95/M13 | P68/M14 | P4/M2 | | | P98/M15 |
| 16 | | P95/M13 | P68/M14 | P4/M2 | | P15/M3 | P98/M15 |
| 17 | P141/M10 | P95/M13 | P68/M14 | P4/M2 | | P15/M3 | P98/M15 |
| 18 | P141/M10 | P95/M13 | P68/M14 | P44/M6 | P75/M2 | P15/M3 | P98/M15 |
| 19 | P141/M10 | | | P44/M6 | P75/M2 | | P98/M15 |
| 20 | P141/M10 | | | | | | P98/M15 |

TABLA 7.17. TABLA OPERATORIA IR150_15M PARA EL DÍA JUEVES

| t | Jueves | | | | | | |
|----|----------|----------|----------|---------|---------|--------|--------|
| | Sala 1 | Sala 2 | Sala 3 | Sala 4 | Sala 5 | Sala 6 | Sala 7 |
| 1 | P136/M11 | | | | | P12/M7 | |
| 2 | P136/M11 | | | | | P12/M7 | |
| 3 | | P85/M14 | | | | P12/M7 | |
| 4 | | P85/M14 | | P145/M7 | | | |
| 5 | P129/M11 | P85/M14 | | P145/M7 | | | |
| 6 | P129/M11 | | | P145/M7 | | | |
| 7 | P129/M11 | | | | | | |
| 8 | P109/M11 | | | | P55/M8 | | |
| 9 | P109/M11 | | P111/M14 | P7/M5 | P55/M8 | | |
| 10 | P109/M11 | | P111/M14 | P7/M5 | P55/M8 | | |
| 11 | P78/M11 | | P111/M14 | P7/M5 | P1/M1 | | |
| 12 | P109/M11 | | P111/M14 | | P1/M1 | | |
| 13 | P62/M11 | | | P124/M7 | | | |
| 14 | P62/M11 | | | P124/M7 | P41/M6 | | |
| 15 | | | | | P41/M6 | | |
| 16 | P114/M11 | | | P36/M5 | P41/M6 | | |
| 17 | P114/M11 | P133/M14 | | P36/M5 | | | |
| 18 | | P133/M14 | | P36/M5 | P93/M11 | P59/M4 | |
| 19 | | P133/M14 | | P144/M1 | P93/M11 | P59/M4 | |
| 20 | | P133/M14 | | P144/M1 | | P59/M4 | |

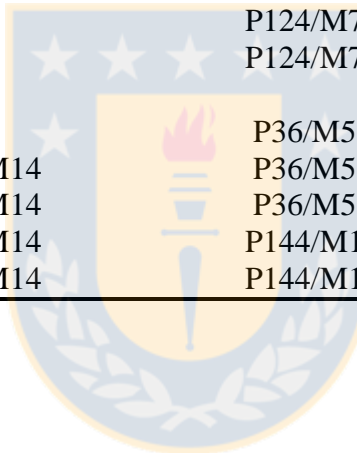


TABLA 7.18. TABLA OPERATORIA IR150_15M PARA EL DÍA VIERNES

| t | Viernes | | | | | | |
|----|---------|----------|---------|---------|--------|--------|----------|
| | Sala 1 | Sala 2 | Sala 3 | Sala 4 | Sala 5 | Sala 6 | Sala 7 |
| 1 | | | | | | | P63/M15 |
| 2 | P67/M10 | | | | P12/M7 | | P63/M15 |
| 3 | P67/M10 | P49/M12 | | | P12/M7 | | P63/M15 |
| 4 | P67/M10 | P49/M12 | | P149/M2 | | P72/M6 | P63/M15 |
| 5 | P67/M10 | P49/M12 | | P149/M2 | | P72/M6 | P63/M15 |
| 6 | P67/M10 | P49/M12 | | | | P72/M6 | P63/M15 |
| 7 | P22/M10 | P49/M12 | | P9/M6 | | P97/M4 | |
| 8 | P22/M10 | P49/M12 | | P9/M6 | | P97/M4 | |
| 9 | P22/M10 | | | P9/M6 | | P97/M4 | |
| 10 | P52/M11 | | | | | | P102/M15 |
| 11 | P52/M11 | | | | | | P102/M15 |
| 12 | P23/M10 | P110/M12 | | | | | P102/M15 |
| 13 | P23/M10 | P110/M12 | P61/M13 | | | | P102/M15 |
| 14 | P23/M10 | P110/M12 | P61/M13 | | | | P102/M15 |
| 15 | P80/M10 | P110/M12 | P61/M13 | | | | P102/M15 |
| 16 | P80/M10 | P137/M12 | P61/M13 | | | | |
| 17 | P80/M10 | P137/M12 | P61/M13 | | | | |
| 18 | P80/M10 | P137/M12 | | | | | |
| 19 | P80/M10 | P137/M12 | | | | | |
| 20 | | P137/M12 | | | | | |

7.4.3 Utilización de salas quirúrgicas para instancias IR100_15M a IR250_22M

La utilización de las siete salas quirúrgicas para las instancias IR100_15M a IR250_22M se muestra en la Tabla 8.19. Además se observa el total de la utilización de cada instancia y se marca cada utilización mayor para cada instancia.

TABLA 7.19. UTILIZACIÓN SALAS QUIRÚRGICAS INSTANCIAS IR100_15M A IR250_22M

| Sala | % Util. IR100_15M | % Util. IR150_15M | % Util. IR150_17M | % Util. IR175_17M | % Util. IR175_18M | % Util. IR175_20M | % Util. IR200_20M | % Util. IR225_20M | % Util. IR225_22M | % Util. IR250_22M |
|-------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| 1 | 51.25 | 65.00 | 70.00 | 72.50 | 80.00 | 76.25 | 91.25 | 92.50 | 81.25 | 87.50 |
| 2 | 41.25 | 52.50 | 71.25 | 61.25 | 73.75 | 62.50 | 81.25 | 85.00 | 86.25 | 78.75 |
| 3 | 51.25 | 50.00 | 48.75 | 61.25 | 58.75 | 70.00 | 83.75 | 70.00 | 86.25 | 81.25 |
| 4 | 36.25 | 61.25 | 63.75 | 68.75 | 66.25 | 68.75 | 78.75 | 83.75 | 76.25 | 81.25 |
| 5 | 53.75 | 63.75 | 53.75 | 71.25 | 60.00 | 67.50 | 70.00 | 62.50 | 83.75 | 81.25 |
| 6 | 40.00 | 51.25 | 51.25 | 58.75 | 62.50 | 66.25 | 70.00 | 82.50 | 77.50 | 86.25 |
| 7 | 46.25 | 36.25 | 63.75 | 55.00 | 45.00 | 43.75 | 67.50 | 73.75 | 77.50 | 76.25 |
| Total | 45.71 | 54.29 | 60.36 | 64.11 | 63.75 | 65.00 | 77.50 | 78.57 | 81.25 | 81.79 |



Capítulo 8. Conclusiones y trabajo futuro

En la actualidad, la programación de cirugías electivas en los hospitales chilenos se realiza en gran parte de forma manual. Esta programación de cirugías electivas, se ayuda en muchos casos con *software* que permiten reunir y ordenar la información de cada cirugía. Sin embargo, dichos *software* no prometen el aumento en el desempeño de toda la unidad de pabellón.

La programación de cirugías, requiere sin dudas, una gran coordinación y trabajo entre todos los servicios de apoyo y relacionados. Además, existe una gran responsabilidad en el manejo de la lista de espera de pacientes en el que es necesaria, claramente, la experiencia del personal médico. Este problema, tal como es mencionado en este informe, es un problema complejo y difícil de modelar, ya que presenta muchos tipos de restricciones tales como: disponibilidad de salas quirúrgicas y médicos, compatibilidad de salas quirúrgicas y médicos con los pacientes en lista de espera, capacidad de camas de hospitalización, disponibilidad de enfermeras y anestesistas, etc. Pero se puede crear una buena aproximación a la realidad del problema, la cual permita facilitar la programación de cirugías.

El uso de metaheurísticas en este tipo de problemas permite encontrar una solución en periodos de tiempo razonables en comparación con los métodos de resolución exactos. El algoritmo GRASP posee un buen desempeño en éste tipo de problemas, logrando tiempos de ejecución relativamente bajos para instancias realistas, aunque este puede ser mejorado en términos de implementación. Sin embargo, *solvers* comerciales como CPLEX poseen un buen desempeño en problemas como este, incluso logrando tiempos razonables en instancias de mayor tamaño.

En comparación con el modelo de programación lineal entero, el algoritmo GRASP se comporta en cierta manera mejor que el modelo IP, por ejemplo en algunas de las instancias ajustadas a la realidad el modelo IP no encuentra solución en tiempos menores a 4 horas, en cambio el algoritmo GRASP encuentra solución en todas ellas, siendo un tiempo máximo de ejecución 8.13 segundos para la instancia de mayor tamaño.

Para la instancia con datos reales de un hospital público chileno utilizada en (Cartes,

2014) el algoritmo GRASP realiza una programación similar al modelo IP, con una deficiencia en el número de sobretiempos utilizados, pero, entregando la solución en una cantidad de tiempo considerablemente menor, siendo el modelo IP resuelto con GLPK en 130800 segundos, versus 1.99 segundos del algoritmo GRASP.

Y finalmente para las instancias reales, se observa que el algoritmo tiene un mejor desempeño con instancias mayores (mayor cantidad de médicos y pacientes), logrando mejores valores en la *fitness* y mayor utilización de las salas quirúrgicas, alcanzando un promedio de 81.79 % de utilización.

En general, la resolución del problema de programación de cirugías electivas con el algoritmo propuesto entregó resultados en una cantidad de tiempo considerablemente menor que el modelo IP resuelto con GLPK, pero aun así los 4728,88 segundos para la instancia real de mayor tamaño no son considerados dentro del tiempo razonable para obtener una programación de cirugías electivas (10-15 minutos).

Utilizar la metaheurística GRASP para resolver el problema de programación de cirugías electivas y asignación de médicos y salas quirúrgicas, la cual no ha sido utilizada según la bibliografía revisada, a pesar del gran número de aplicaciones de la metaheurística, entrega una fuerte base para investigación del problema bajo este tipo de algoritmo, el cual GRASP puede ser mejorado en términos de eficiencia en los vecindarios propuestos y además con una interfaz que permita al usuario un uso más amigable con el algoritmo.

Bibliografía

- Agnētis, A., Coppi, A., Corsini, M., Dellino, G., Meloni, C., & Pranzo, M. (2012). Long term evaluation of operating theater planning policies. *Operations Research for Health Care*, 1(4), 95–104. <http://doi.org/10.1016/j.orhc.2012.10.001>
- Augusto, V., Xie, X., & Perdomo, V. (2010). Operating theatre scheduling with patient recovery in both operating rooms and recovery beds. *Computers & Industrial Engineering*, 58(2), 231–238. <http://doi.org/10.1016/j.cie.2009.04.019>
- Burkard, R. E., Dell'Amico, M., & Martello, S. (2009). *Assignment Problems*. SIAM.
- Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201(3), 921–932. <http://doi.org/10.1016/j.ejor.2009.04.011>
- Delorme, X., Gandibleux, X., & Rodriguez, J. (2004). GRASP for set packing problems. *European Journal of Operational Research*, 153(3), 564–580. [http://doi.org/10.1016/S0377-2217\(03\)00263-7](http://doi.org/10.1016/S0377-2217(03)00263-7)
- Denton, B., Viapiano, J., & Vogl, A. (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science*, 10(1), 13–24. <http://doi.org/10.1007/s10729-006-9005-4>
- Fei, H., Meskens, N., & Chu, C. (2010). A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers & Industrial Engineering*, 58(2), 221–230. <http://doi.org/10.1016/j.cie.2009.02.012>

- Fügener, A., Hans, E. W., Kolisch, R., Kortbeek, N., & Vanberkel, P. T. (2014). Master surgery scheduling with consideration of multiple downstream units. *European Journal of Operational Research*, 239(1), 227–236. <http://doi.org/10.1016/j.ejor.2014.05.009>
- Gogna, A., & Tayal, A. (2013). Metaheuristics: review and application. *Journal of Experimental & Theoretical Artificial Intelligence*, 25(4), 503–526. <http://doi.org/10.1080/0952813X.2013.782347>
- Gordon, T., Paul, S., Lyles, A., & Fountain, J. (1988). Surgical unit time utilization review: Resource utilization and management implications. *Journal of Medical Systems*, 12(3), 169–179. <http://doi.org/10.1007/BF00996639>
- Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. H. G. R. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. En E. L. J. and B. H. K. P.L. Hammer (Ed.), *Annals of Discrete Mathematics* (Vol. Volume 5, pp. 287–326). Elsevier. Recuperado a partir de <http://www.sciencedirect.com/science/article/pii/S016750600870356X>
- Guerriero, F., & Guido, R. (2011). Operational research in the management of the operating theatre: a survey. *Health Care Management Science*, 14(1), 89–114. <http://doi.org/10.1007/s10729-010-9143-6>
- Liu, Y., Chu, C., & Wang, K. (2011). A new heuristic algorithm for the operating room scheduling problem. *Computers & Industrial Engineering*, 61(3), 865–871. <http://doi.org/10.1016/j.cie.2011.05.020>

- Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6), 369–395. <http://doi.org/10.1007/s00158-003-0368-6>
- Meskens, N., Duvivier, D., & Hanset, A. (2013). Multi-objective operating room scheduling considering desiderata of the surgical team. *Decision Support Systems*, 55(2), 650–659. <http://doi.org/10.1016/j.dss.2012.10.019>
- Mullen, P. M. (2003). Prioritising waiting lists: how and why? *European Journal of Operational Research*, 150(1), 32–45. [http://doi.org/10.1016/S0377-2217\(02\)00779-8](http://doi.org/10.1016/S0377-2217(02)00779-8)
- Park, C., & Seo, J. (2012). A GRASP approach to transporter scheduling and routing at a shipyard. *Computers & Industrial Engineering*, 63(2), 390–399. <http://doi.org/10.1016/j.cie.2012.04.010>
- Pinedo, M. (2008). *Scheduling: Theory, Algorithms, and Systems*. Springer.
- Pradenas, L., Vidal, F., Parada, L., & Melgarejo, E. (2012). An algorithm to surgery scheduling and surgeons assignment in a public hospital. *Revista del Instituto Chileno de Investigación de Operaciones*, 2(1), 20–29.
- Pradenas Rojas, L., & Matamala Vergara, E. (2012). Una formulación matemática y de solución para programar cirugías con restricciones de recursos humanos en el hospital público. *Ingeniare. Revista chilena de ingeniería*, 20(2), 230–241. <http://doi.org/10.4067/S0718-33052012000200010>
- Resende, M., & Ribeiro, C. (2003). Greedy randomized adaptive search procedures. En *Handbook of metaheuristics* (pp. 219–249). Springer New York.

- Riise, A., & Burke, E. K. (2011). Local search for the surgery admission planning problem. *Journal of Heuristics*, 17(4), 389–414. <http://doi.org/10.1007/s10732-010-9139-x>
- Roland, B., Di Martinelly, C., Riane, F., & Pochet, Y. (2010). Scheduling an operating theatre under human resource constraints. *Computers & Industrial Engineering*, 58(2), 212–220. <http://doi.org/10.1016/j.cie.2009.01.005>
- Ropke, S. (2005). Heuristic and exact algorithms for vehicle routing problems. *Unpublished PhD thesis, Computer Science Department, University of Copenhagen.*
- Santibáñez, P., Begen, M., & Atkins, D. (2007). Surgical block scheduling in a system of hospitals: an application to resource and wait list management in a British Columbia health authority. *Health Care Management Science*, 10(3), 269–282. <http://doi.org/10.1007/s10729-007-9019-6>
- Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. John Wiley & Sons.
- Testi, A., Tanfani, E., Valente, R., Ansaldo, G. L., & Torre, G. C. (2008). Prioritizing surgical waiting lists. *Journal of Evaluation in Clinical Practice*, 14(1), 59–64. <http://doi.org/10.1111/j.1365-2753.2007.00794.x>
- Vijay V. Vazirani. (2001). *Approximation Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA.
- Vijayakumar, B., Parikh, P. J., Scott, R., Barnes, A., & Gallimore, J. (2013). A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *European Journal of Operational Research*, 224(3), 583–591. <http://doi.org/10.1016/j.ejor.2012.09.010>

Wolff, P., Durán, G., Rey, P. (2012). Modelos de programación matemática para asignación de pabellones quirúrgicos en hospitales públicos. *Revista Ingeniería de Sistemas*, 26, pp. 23-48.

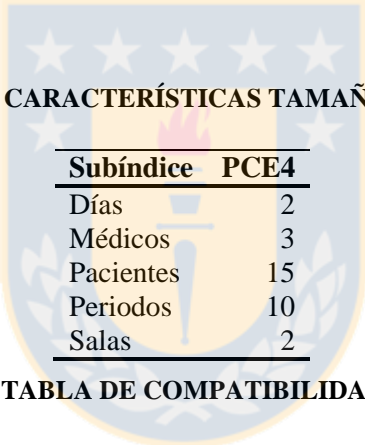


Anexo A. Instancias

A.1. Instancia PCE 4

La instancia PCE 4 utilizada en (Cartes, 2014) es la instancia ocupada para la parametrización del algoritmo GRASP. Donde se detalla en la Tabla A.1 el tamaño de la instancia, en la Tabla A.2 la tabla de compatibilidad que muestra; la ID del paciente, prioridad, duración de cirugía, y la compatibilidad sala-paciente y médico-paciente, la cual indica que un médico o sala es compatible con un paciente si es que tiene un 1 en la celda respectiva o 0 en otro caso. Las Tablas A.3 y A.4 muestra la disponibilidad horaria de las salas y médicos para el día 1 y 2 respectivamente. Finalmente la Tabla A.5 muestra el costo de utilización de las salas en sobretiempo.

TABLA A.1. CARACTERÍSTICAS TAMAÑO INSTANCIA PCE 4



| Subíndice | PCE4 |
|------------------|-------------|
| Días | 2 |
| Médicos | 3 |
| Pacientes | 15 |
| Periodos | 10 |
| Salas | 2 |

TABLA A.2 TABLA DE COMPATIBILIDAD, INSTANCIA PCE4

| ID Paciente | Prioridad | Duración Cirugía (t) | Sala 1 | Sala 2 | Med 1 | Med 2 | Med 3 |
|--------------------|------------------|-----------------------------|---------------|---------------|--------------|--------------|--------------|
| 1 | 0.92 | 1 | 1 | 1 | 1 | 0 | 1 |
| 2 | 0.52 | 1 | 1 | 0 | 1 | 0 | 1 |
| 3 | 0.22 | 2 | 1 | 1 | 1 | 0 | 0 |
| 4 | 0.99 | 2 | 0 | 0 | 0 | 1 | 1 |
| 5 | 0.36 | 3 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0.24 | 3 | 1 | 1 | 0 | 1 | 1 |
| 7 | 0.96 | 1 | 1 | 1 | 0 | 1 | 1 |
| 8 | 0.24 | 2 | 1 | 1 | 0 | 0 | 1 |
| 9 | 0.95 | 3 | 1 | 0 | 1 | 0 | 1 |
| 10 | 0.71 | 1 | 1 | 0 | 1 | 0 | 0 |
| 11 | 0.49 | 3 | 1 | 0 | 0 | 1 | 0 |
| 12 | 0.65 | 1 | 1 | 0 | 0 | 1 | 1 |
| 13 | 0.81 | 2 | 1 | 1 | 0 | 1 | 0 |
| 14 | 0.17 | 2 | 1 | 1 | 0 | 1 | 1 |
| 15 | 0.97 | 1 | 1 | 1 | 0 | 1 | 0 |

TABLA A.3. TABLA DE DISPONIBILIDAD HORARIA PARA EL DIA 1, INSTANCIA PCE 4

| Periodo (t) | Sala 1 | Sala 2 | Med 1 | Med 2 | Med 3 |
|--------------------|---------------|---------------|--------------|--------------|--------------|
| 1 | 1 | 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 1 | 1 | 0 |
| 3 | 1 | 1 | 1 | 1 | 0 |
| 4 | 1 | 1 | 1 | 1 | 0 |
| 5 | 1 | 1 | 0 | 1 | 1 |
| 6 | 1 | 0 | 0 | 1 | 1 |
| 7 | 1 | 0 | 0 | 1 | 1 |
| 8 | 1 | 0 | 0 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | 1 |

TABLA A.4. TABLA DE DISPONIBILIDAD HORARIA PARA EL DÍA 2, INSTANCIA PCE4

| Periodo (t) | Sala 1 | Sala 2 | Med 1 | Med 2 | Med 3 |
|--------------------|---------------|---------------|--------------|--------------|--------------|
| 1 | 1 | 0 | 1 | 0 | 1 |
| 2 | 1 | 0 | 1 | 0 | 1 |
| 3 | 1 | 0 | 1 | 0 | 1 |
| 4 | 1 | 1 | 1 | 0 | 1 |
| 5 | 1 | 1 | 0 | 0 | 1 |
| 6 | 1 | 1 | 0 | 0 | 0 |
| 7 | 1 | 1 | 0 | 0 | 0 |
| 8 | 1 | 1 | 0 | 0 | 0 |
| 9 | 1 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | 1 |

**TABLA A.5. COSTO UTILIZACIÓN DE SALAS QUIRÚRGICAS EN SOBRETIEMPO,
INSTANCIA PCE4**

| Sala | CST Sala Quirúrgica |
|-------------|----------------------------|
| 1 | 0.15 |
| 2 | 0.22 |

A.2. Instancia PCE 17

La instancia PCE 17 utilizada en (Cartes, 2014) es la instancia de comparación del modelo IP con el algoritmo GRASP. Donde se detalla en la Tabla A.6 el tamaño de la instancia, en la Tabla A.7 la tabla de compatibilidad que muestra; la ID del paciente, prioridad, duración de cirugía, y la compatibilidad sala-paciente y médico-paciente, la cual indica que un médico o sala es compatible con un paciente si es que tiene un 1 en la celda respectiva o 0 en otro caso. Las Tablas A.8 y A.9 muestra la disponibilidad horaria de las salas y médicos para el día 1 y 2 respectivamente. Finalmente, la Tabla A.10 muestra el costo de utilización de las salas en sobretiempo.

TABLA A.6. CARACTERÍSTICAS TAMAÑO DE LA INSTANCIA PCE 17

| Subíndice | PCE17 |
|-----------|-------|
| Días | 2 |
| Médicos | 5 |
| Pacientes | 15 |
| Periodos | 20 |
| Salas | 3 |

TABLA A.7. TABLA DE COMPATIBILIDAD INSTANCIA PCE 17

| ID Pcte. | Prioridad | Tiempo Cir. Estimado (min) | Sala 1 | Sala 2 | Sala 3 | Med 1 | Med 2 | Med 3 | Med 4 | Med 5 |
|----------|-----------|----------------------------|--------|--------|--------|-------|-------|-------|-------|-------|
| 1 | 0.24 | 70 | | | | 1 | 1 | | | |
| 2 | 0.15 | 60 | | | | 1 | 1 | | | |
| 3 | 0.09 | 70 | | | | 1 | 1 | | | |
| 4 | 0.36 | 80 | | | | 1 | | 1 | | |
| 5 | 0.11 | 80 | | | | 1 | | 1 | | |
| 6 | 0.21 | 80 | | | | 1 | | 1 | | |
| 7 | 0.32 | 70 | | | | 1 | | 1 | | |
| 8 | 0.19 | 160 | | 1 | | | | | | 1 |
| 9 | 0.57 | 160 | | 1 | | | | | | 1 |
| 10 | 0.79 | 170 | | 1 | | | | | | 1 |
| 11 | 0.79 | 85 | | | | 1 | | | 1 | |
| 12 | 0.77 | 90 | | | | 1 | | | 1 | |
| 13 | 0.29 | 220 | 1 | | | | | | | 1 |
| 14 | 0.11 | 140 | 1 | | | | | | | 1 |
| 15 | 0.13 | 140 | 1 | | | | | | | 1 |

TABLA A.8 TABLA DE DISPONIBILIDAD HORARIA PARA EL DIA 1, INSTANCIA PCE17

| Periodo (t) | Sala 1 | Sala 2 | Sala 3 | Med 1 | Med 2 | Med 3 | Med4 | Med 5 |
|-------------|--------|--------|--------|-------|-------|-------|------|-------|
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 12 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 13 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 14 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 15 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 16 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 17 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 18 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 19 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 20 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

TABLA A.9. TABLA DE DISPONIBILIDAD HORARIA PARA EL DIA 2, INSTANCIA PCE17

| Periodo (t) | Sala 1 | Sala 2 | Sala 3 | Med 1 | Med 2 | Med 3 | Med 4 | Med 5 |
|-------------|--------|--------|--------|-------|-------|-------|-------|-------|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 11 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 13 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 15 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 16 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 17 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 18 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 19 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 20 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

TABLA A.10. COSTO UTILIZACIÓN DE SALAS QUIRÚRGICAS EN SOBRETIEMPO, INSTANCIA PCE17

| Sala | CST Sala Quirúrgica |
|-------------|----------------------------|
| 1 | 0.17 |
| 2 | 0.26 |
| 3 | 0.44 |

A.3. Instancias reales de mayor tamaño

Las instancias IR100_15M hasta IR250_22M son las instancias utilizadas solo en con el algoritmo GRASP y son formadas en base a la instancia IR250_22M la cual posee 250 pacientes en la lista de espera y 22 médicos. Es decir, esta instancia es la de mayor tamaño y las instancias restantes son una parte de ésta instancia.

La Tabla A.11 detalla el tamaño de la instancia IR250_M. La Tabla A.12 es la tabla de compatibilidad que muestra; la ID del paciente, prioridad, duración de cirugía, y la compatibilidad sala-paciente y médico-paciente, la cual indica que un médico o sala es compatible con un paciente si es que tiene un 1 en la celda respectiva o 0 en otro caso

TABLA A.11. CARACTERÍSTICAS TAMAÑO DE LA INSTANCIA IR250_22M

| Subíndice | IR250_22M |
|------------------|------------------|
| Días | 5 |
| Médicos | 22 |
| Pacientes | 250 |
| Periodos | 20 |
| Salas | 7 |

**TABLA A.12. COSTO UTILIZACIÓN DE SALAS QUIRÚRGICAS EN SOBRETIEMPO,
INSTANCIA IR250_22M**

| Sala | CST Sala Quirúrgica |
|-------------|----------------------------|
| 1 | 0.17 |
| 2 | 0.26 |
| 3 | 0.32 |
| 4 | 0.21 |
| 5 | 0.14 |
| 6 | 0.28 |
| 7 | 0.33 |



TABLA A.13. TABLA DE COMPATIBILIDAD INSTANCIA IR250_22M

| ID | Prio | T | S 1 | S 2 | S 3 | S 4 | S 5 | S 6 | S 7 | M 1 | M 2 | M 3 | M 4 | M 5 | M 6 | M 7 | M 8 | M 9 | M 10 | M 11 | M 12 | M 13 | M 14 | M 15 | M 16 | M 17 | M 18 | M 19 | M 20 | M 21 | M 22 |
|----|------|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1 | 0.18 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | |
| 2 | 0.94 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 3 | 0.65 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 4 | 0.64 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0.66 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0.43 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 7 | 0.54 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0.92 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 9 | 0.41 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 10 | 0.47 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 11 | 0.65 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0.63 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 13 | 0.40 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 0.67 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 0.46 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 16 | 0.28 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 0.85 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 18 | 0.59 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 19 | 0.53 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0.81 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 21 | 0.44 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | 0.30 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 0.85 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 0.61 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 25 | 0.88 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 54 | 0.63 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 55 | 0.81 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 56 | 0.87 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 57 | 0.98 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 58 | 0.26 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 59 | 0.95 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 60 | 0.62 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 61 | 0.55 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 62 | 0.99 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 63 | 0.88 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 64 | 0.56 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 65 | 0.48 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 66 | 0.57 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 67 | 0.84 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 68 | 0.68 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 69 | 0.42 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 70 | 0.17 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 71 | 0.42 | 4 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 72 | 0.92 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 73 | 0.64 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 74 | 0.74 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 75 | 0.13 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 76 | 0.13 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 77 | 0.22 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 78 | 0.18 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 79 | 0.31 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 80 | 0.98 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 81 | 0.36 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 82 | 0.75 | 4 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 83 | 0.42 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 84 | 0.11 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 85 | 0.85 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 86 | 0.19 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 87 | 0.27 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 88 | 0.58 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 89 | 0.33 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 90 | 0.32 | 4 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 91 | 0.58 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 92 | 0.45 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 93 | 0.87 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 94 | 0.97 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 95 | 0.47 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 96 | 0.44 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 97 | 0.93 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 98 | 0.73 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 99 | 0.81 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 100 | 0.66 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 101 | 0.19 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 102 | 0.88 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 103 | 0.93 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 104 | 0.46 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 105 | 0.30 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 106 | 0.11 | 6 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 107 | 0.91 | 6 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 108 | 0.31 | 6 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 109 | 0.12 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 110 | 0.98 | 4 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 111 | 0.85 | 4 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 112 | 0.18 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 113 | 0.69 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 114 | 0.77 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 115 | 0.13 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 116 | 0.85 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 117 | 0.63 | 4 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 118 | 0.18 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 119 | 0.64 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 120 | 0.66 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 121 | 0.36 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 122 | 0.62 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 123 | 0.10 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 124 | 0.69 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 125 | 0.10 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 126 | 0.28 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 127 | 0.15 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 128 | 0.40 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 129 | 0.83 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 130 | 0.92 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 131 | 0.46 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 132 | 0.16 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 133 | 0.27 | 4 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 134 | 0.34 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 135 | 0.77 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 136 | 0.33 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 137 | 0.76 | 5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 166 | 0.17 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 167 | 0.95 | 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 168 | 0.32 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 169 | 0.79 | 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 170 | 0.24 | 5 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | |
| 171 | 0.94 | 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 172 | 0.47 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 173 | 0.23 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 174 | 0.32 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 175 | 0.85 | 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 176 | 0.14 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 177 | 0.88 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 178 | 0.46 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 179 | 0.91 | 7 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 180 | 0.43 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 181 | 0.22 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 182 | 0.37 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 183 | 0.78 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 184 | 0.31 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 185 | 0.19 | 2 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 186 | 0.59 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 187 | 0.20 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 188 | 0.79 | 6 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 189 | 0.88 | 6 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 190 | 0.44 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 191 | 0.78 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 192 | 0.66 | 4 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 193 | 0.74 | 4 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 222 | 0.49 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 223 | 0.98 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 224 | 0.31 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 225 | 0.22 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 226 | 0.75 | 4 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 227 | 0.83 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 228 | 0.71 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 229 | 0.15 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 230 | 0.53 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 231 | 0.49 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 232 | 0.84 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 233 | 0.51 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 234 | 0.39 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 235 | 0.78 | 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 236 | 0.39 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 237 | 0.68 | 6 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 238 | 0.38 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 239 | 0.19 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 240 | 0.30 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 241 | 0.91 | 5 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 242 | 0.60 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 243 | 0.21 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 244 | 0.53 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 245 | 0.61 | 4 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 246 | 0.80 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 247 | 0.22 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 248 | 0.77 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 249 | 0.32 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

250 | 0.73 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0

